IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Salim et al.	)
		) Art Unit: Not Yet
Serial No.:	09/765,964	) Assigned
		) Examiner:
Filing Date:	January 19, 2001	) Not Yet Assigned
		)
Title:	METHOD AND APPARATUS FOR IMPLEMENTING	)
	AN ACTIVE INFORMATION MODEL	)
		)
Att'y File No.:	19382-004	)

PETITION TO BE ACCORDED FILING DATE UNDER 37 C.F.R. 1.53(e)

To: Assistant Commissioner of Patents
      Washington, DC 20231

Sir:

In response to the Notice of Incomplete Nonprovisional Application, dated March 27, 2001, Applicants respectfully petition under 37 C.F.R. § 1.53(e) that the above-identified patent application be accorded the filing date of January 19, 2001, and state as follows:

Facts and Relief Requested

1.   An Express Mail package containing this patent application was mailed to the Patent and Trademark Office on January 19, 2001, enclosing the following items:

     (1)   Patent Application, including ninety (90) pages of specifications, one (1) page of claims (1 claim), and Abstract, entitled METHOD AND APPARATUS FOR IMPLEMENTING AN ACTIVE INFORMATION MODEL;

     (2)   Forty (40) sheets of drawings (Figures 1-38);

     (3)   Attachment A, Provisional patent application entitled "DATASOURCE HARMONIZER (27 sheets);

(4)   Attachment B, "harmeng: Logical View Report" (26 sheets);

(5)   Attachment C, "References" (1 sheet);

(6)   Transmittal Letter to Assistant Commissioner of Patents; and

(7)   Return Postcard.

2.   Applicants subsequently received the return postcard from the Patent and Trademark Office, reflecting actual receipt and according Serial No. 09/765964, and a Filing Date of 01/19/01. The postcard also states that "Forty (40) sheets of drawings (Figures 1-38)" were enclosed with the patent application.

3.   A "Notice of Incomplete Nonprovisional Application", dated March 27, 2001, has been received by Applicants which stated, "The application was deposited without drawings. . . . The filing date of this nonprovisional application will be the date of receipt of the [required] items."

4.   Because every indication is that the Drawings were submitted with the originally filed patent application, and because the returned postcard indicates that they were actually received by the Patent and Trademark Office, Applicants respectfully request that the patent application be accorded the filing date of January 19, 2001.

<div align="center">First Alternative Basis</div>

5.   In the alternative, and if the January 19, 2001 filing date is not granted as requested in paragraphs 1 - 4 above, Applicants request that the enclosed drawings be added to the application by amendment, and because the enclosed drawings are described in the specification and so do not constitute new matter, that the application be accorded the filing date of January 19, 2001.

Second Alternative Basis

6.   In the alternative, and if the January 19, 2001 filing date is not granted as requested in either paragraphs 1 - 4 above, or as requested in the first alternative basis set forth in paragraph 5 above, Applicants request that the enclosed drawings be accepted, and that the application be accorded at least the filing date which is the mailing date of this paper. Applicants make this second alternative request solely for the purpose of having at least such filing date as is the mailing date hereof, and without prejudice to pursuing the other bases set forth in paragraphs 1 - 5 herein, including by way of further petition or other appropriate avenue.

Attachments

In support of the petition, Applicants are submitting as attachments hereto, a full copy of the patent application, as originally filed, along with a copy of the returned receipt postcard.

Fee Statement

Since this appears to be Patent Office error, Applicants believe no fee is necessary. But if any fees are required by this paper, the undersigned hereby authorizes any such fees to be charged to Deposit Account 03-1725.

Respectfully submitted,

Dated: May 18, 2001

Thomas C. Folsom, Reg. No. 35,514
CHRISMAN, BYNUM & JOHNSON, P.C.
1900 Fifteenth Street
Boulder, CO 80302
(303) 546-1300
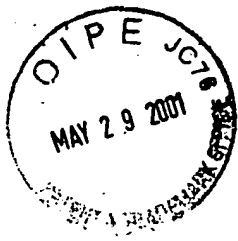
DATASOURCE HARMONIZER

INVENTED BY:

Mohammad S. Salim,

Barbara J. Rossner,

and Ronald M. Barber

DATASOURCE HARMONIZER

BACKGROUND OF INVENTION

1.   Field of the Invention

This invention relates to processes governing the implementation of a collaborative computing environment within an enterprise and, more specifically, to a core software component (the Invention) which makes it possible to implement this environment. The result of deploying the Invention is a virtual, interactive, information management system that is not uniquely dependent on any server or any specific apparatus for its operation. The implementation of the system can use a variety of electronic data processing devices to assist the users' interaction with the system and data storage serves and communication links to provide for storage and transmission of persistent information. Electronic devices as identified above may be workstations, laptop computers, and hand held data entry and communication devices. The choice of these devices is based on the needs of the users of the system and the system itself is not dependent on any apparatus.

2.   Description of the Related Art

Building an information system for specific business or other enterprises is an important and difficult problem. The following discussion will stress building an information system for the health care industry and its enterprises for HomeCare, a home care agency[1], as an example in describing the problem that is addressed by the Invention in the interest of clarity, but not as a limitation, and because of the growth of this industry in recent years. Industry sources report the current market for healthcare information technology in the United States to be $17.3 billion in 1997 and project the market to grow to $27.9 billion by 2002. Of that market, the second highest IT priority for 1998 is integrating (connecting) systems in a multi-vendor environment (second only to recruiting and retaining qualified IT staff; as reported in the Ninth Annual HIMSS Leadership Survey).

The problems in this industry are a result of a complex shift in needs in the healthcare community. First, the level of sophistication in the practice of information management is rising. Specifically:

- Healthcare professionals are shifting away from simply operating vendor-supplied application programs. These professionals are attempting to select or even implement information systems that incorporate their own policies and practices.

- Powerful desktop computers, high bandwidth networks, and the explosion in the use of the Internet have created a situation where distributed computing is commonplace.

- The recent release of healthcare industry information exchange standards such as HL7, DICOM, CCOW, and XML have resulted in a language set that makes it possible to create healthcare translators.

Secondly, the technology base that governs the implementation of existing solutions has proven to be inadequate to meet the increased demand on effective information management, for example:

- The creation, usage, and management of information have been severely hampered by the number of proprietary data protocols presently in use in the healthcare industry.

---

[1] We are using a regulated agency as an example because the processes governing the operation of such an organization are clearly defined and well documented. We are assuming that HomeCare is conformant to the appropriate regulations therefore avoiding a possibly long discussion relating to its operational maturity.

- The increasing pressure by payor networks and government regulations for cost reductions has created an extreme need for caregivers and administrators to have access to high quality, timely information.

- Software systems presently offered to the healthcare industry are not user configurable and disrupt the workflow of the business. For caregivers who are most pressed for time and therefore resistant to change, adoption of these systems is difficult at best case and totally ineffective in most cases.

The mission of HomeCare is: "to provide a comprehensive range of home care services to the community and strive for the highest level of professional and technical competence in those services".

Figure 1 provides an organizational view of an information system for HomeCare. Activities in this organization are substantially influenced by two external sources: the Doctors and the Payers. Additionally, regulatory considerations impose requirements from the perspective of quality and longevity of the patient records. Furthermore, professional and business considerations impose requirements about timeliness of services and preparation of information relative to the services. Caregivers deliver the actual services that HomeCare offers according to methods and policies that are endorsed by the Caregiver community in general and that are modified according to the judgement of the professionals who are employed by HomeCare.
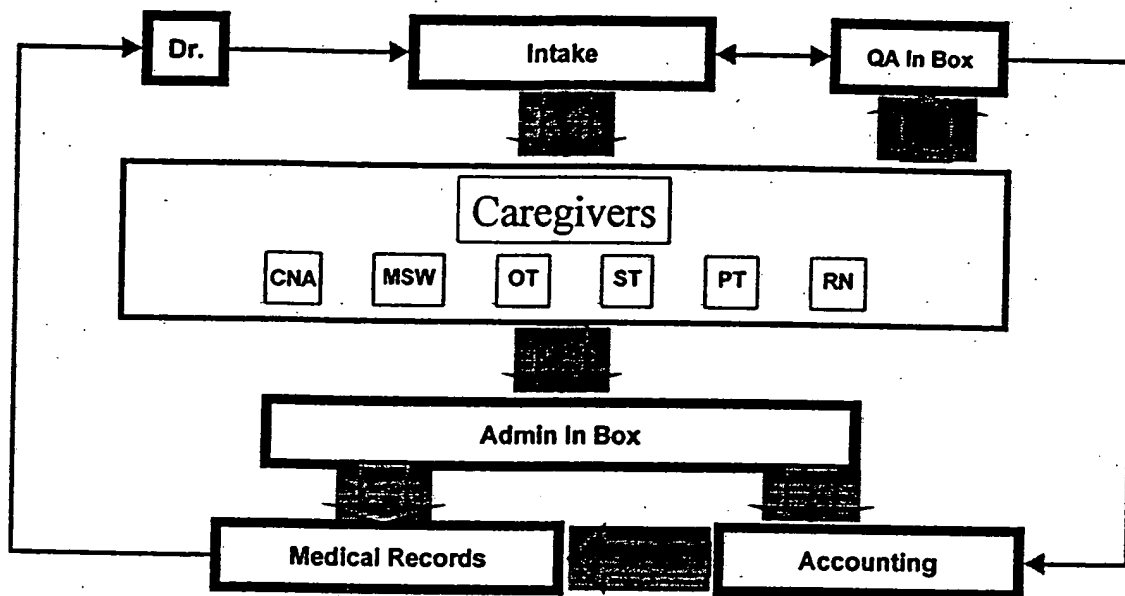

Figure 1. Organizational view of an Information system for a home care agency

Within this organization, policies regarding formal flow of information are based on:

- Agreements among individuals and groups,

- Job assignments,

- Government regulations,

- Policies of payers and Doctors or client hospitals.

-

It is important to recognize that these policies are NOT based on implementation considerations of the electronic data processing and communication equipment.

Moving toward automation: The traditional approach

However, management understands that proper use of state of the art technology in information management can improve quality and reduce operating cost. Management has identified an important focus area for process improvement as automation. It has specified a goal to: "*develop a strategic plan and budget to support the automation of core business processes including tactical planning for interim computer operations, and the evaluation of appropriate available computer technology in moving toward an electronic patient record and total electronic bill generation*".

The phrase "appropriate available computer technology" represents an element of the operational processes that are not under the control of any of the factors identified above. Given the above statement of objectives and operational realities, it seems inevitable that the organization will move toward higher degrees of automation in phases. A traditional decision process to prioritize the goals of each phase of automation will utilize a decision support model[2] that is shown in Figure 2.

Mission: Home Care

Requirements for each phase of automation process

Computer and workstations

Need to automate

Available technology

Policies governing allocation of resources to development of information system

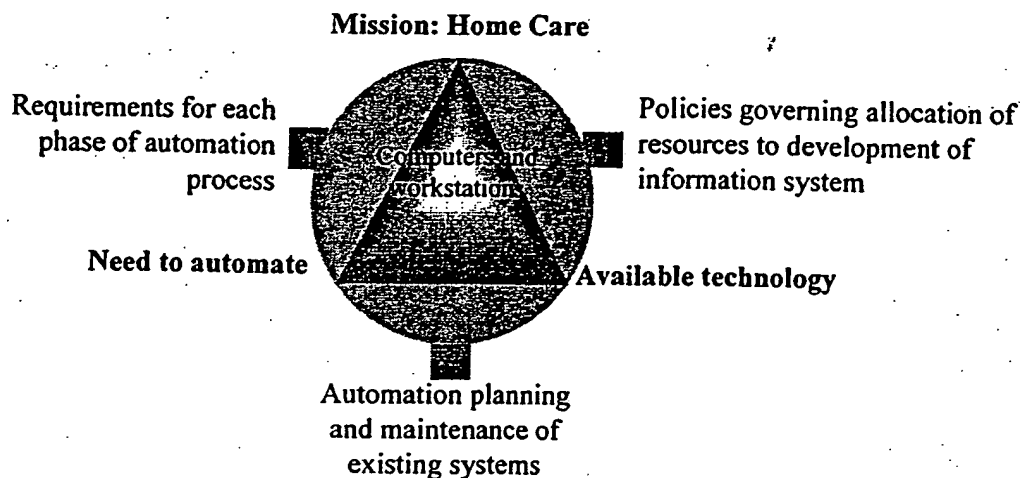Automation planning and maintenance of existing systems

Figure 2. Decision support model based on state of technology

This model identifies the elements of the decision process and highlights the relationships between these elements. It demonstrates the following principals:

1. The overall objective is to improve the effectiveness of the organization toward performance of its mission, which is to provide care for patients at their home.

2. The organization's mission and the need to automate some portion of the business drive requirements for each phase of automation.

3. The specific goals of each phase of automation will be determined by the individuals or groups who are impacted (expressing or resisting the need to automate) and the state of tools and services that are affordable to the organization (available technology).

4. Policies that reflect the organization's mission as well as the available technologies govern the allocation of resources to each phase of automation.

This is a commonly used model and somewhat intuitive from the perspective of those who supply services and technology to this organization. However this model is biased toward technological considerations at the expense of organization's need to plan and execute its long-term goals and decisions based on this model tend to create conflict between the users and service providers.

---

[2] This decision support model is part of inventors' proprietary product development framework. It is an element of the patent application: "myPIP: A framework for a user centric system for storage, retrieval, management and communication of diverse information at a global level."

A frequent outcome of using this model to support of automation decisions is to limit the practices of various segments of the organization to the capabilities of the most recently installed component of automation. In this way, the organization can reap the benefits and amortize the cost of each investment before committing to further enhancements. It may take the organization 6-12 months to adopt an automation component, and the nature of HomeCare's business that forces it to use only proven products. Phased implementation further exasperates this situation by forcing the projects to execute in a sequential manner. These facts practically guarantee that the HomeCare is using obsolete products at all times and in all departments.

Traditional implementations of information systems fail to support this approach because they are based on the architecture of one or more information-processing engine (computers and workstations) as the central component of the information management system. The invention that is being described in these notes makes it possible to put the needs of the users of the information system in the center of the decision support process because it facilitates creation and maintenance of an active Information Model.

Vendors of integration and connectivity engines, such as HIE, STC or TSI, have adapted solutions to the healthcare IS market that were originally designed for financial computing. Impetus for development of these products has been mainframe computing, a characteristic that is reflected in their functionality, packaging, and cost structure. The problems of clinical healthcare, however, include image and records management and operational protocols, as well as data communication protocols and formats. These problems are not addressed by those solutions, making those solutions inappropriate for this market segment.

SUMMARY OF THE INVENTION

Objectives:

The methods and systems of the invention are designed to meet the business needs of enterprises, specifically those of healthcare providers. In this regard, the objectives of the method and system of the invention include the following:
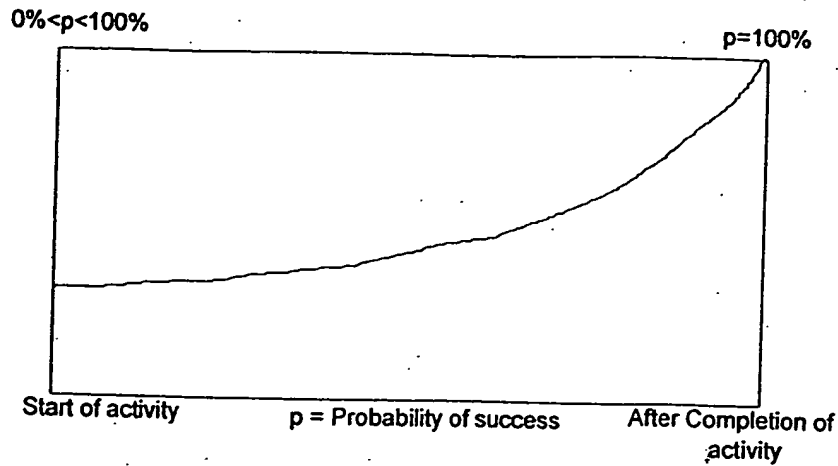
- Information systems and applications must be easy to purchase and provide an immediate ROI.

- They must integrate easily into the existing environment and be easy to use.

- They should be configurable to allow the business to incorporate its unique processes. "I want it my way, not the way a programmer thinks I want it" is a phrase we have heard over and over during our market research interviews.

- They should be compliant with current and immerging medical communication and reporting standards (i.e. ASTM, IEEE P1073, DICOM, HL7, OASIS and MDS).

- They should be flexible, enabling access to a wide variety of databases and data sources and possess the ability to change as business processes change.

- Information management solutions must be scalable to allow the business to start small, yet maintain the ability to grow as business needs dictate.

- Technology must be supported, if not embraced, by leading healthcare IS professionals (i.e. consultants and systems integrators) to allow healthcare providers to enlist knowledgeable support when they are consolidating or downsizing their operations.

With these objectives in mind, the result of deploying the Invention is a virtual, interactive, information management system that is not uniquely dependent on any server or any specific apparatus for its operation. The implementation of the system can use a variety of electronic data processing devices to assist the users' interaction with the system and data storage serves and communication links to provide for storage and transmission of persistent information. Electronic devices as identified above may be workstations, laptop computers, and hand held data entry and communication devices. The choice of these devices is based on the needs of the users of the system and the system itself is not dependent on any apparatus. The primary governing factors in the information system that deploys the Invention are the processes that the user community has adopted in contrast to a traditional information system that imposes processes that facilitate use of its components.

Definition of Process

1: a particular course of action intended to achieve a results; "the procedure of obtaining a driver's license"; "it was a process of trial and error" [syn: procedure]

2: a sustained phenomenon or one marked by gradual changes; "events now in process"; "the process of calcification begins later for boys than for girls"

3: the performance of some composite cognitive activity; an operation that affects mental contents; "the process of thinking"; "the act of remembering" [syn: cognitive process, operation, cognitive operation, act]

4: a writ issued by authority of law; usually compels the defendant's attendance in a civil suit; failure to appear results in a default judgment against the defendant [syn: summons]

5: a mental process that you are not directly aware of; "the process of denial" [syn: unconscious process]

6: a natural prolongation or projection from a part of an organism either animal or plant; "a bony process" [syn: outgrowth, appendage] v 1: deal with in a routine way: "I'll handle that one"; "process a loan"; "process the applicants" 2: subject to a process or treatment, often with the aim of readying for some purpose; "process cheese"; "process hair"; "process water" [syn: treat] 3: perform mathematical and logical operations on (data) according to programmed instructions in order to obtain the required information; "The results of the elections were still being processed when he gave his acceptance speech" 4: institute legal proceedings against; file a suit against; "He was warned that the district attorney would process him" [syn: sue, litigate] 5: shape, form, or improve something: "work stone into tools"; "process iron" [syn: work, work on] 6: serve somebody with a warrant or summons; "He was processed by the sheriff" [syn: serve, swear out]

7: march in a procession; "They processed into the dining room" [syn: march]
Source: WordNet ® 1.6. © 1997 Princeton University

The concept of "process" as used in the context of The Invention indicates a continuous focus by the participants on achieving the intended results. Specifically, and for the purposes of information management, a process is the formal element for managing the probability of success during the course of an activity.

0%<p<100%   p=100%

Start of activity   p = Probability of success   After Completion of
                                                            activity

There is a chance that any activity that is undertaken by an organization may fail i.e. probability of success for that activity is less than 100%. Once the activity is completed and the expected results are achieved the probability of success is 100%. A process is the thoughtful and formal method that is adopted by the individual or organization for systematically improving the probability of success from its starting value to the success outcome.

The active elements of any process in an organization are the individuals who participate in the process. Within the framework of the governing process, these individuals evaluate progress, and redefine and execute the specific actions that result in improving the probability of success. An information system that serves such a process must facilitate this redefinition and execution of these specific tasks. It follows that a critical attribute of the Invention is its ability to model an existing process into a coherent, active information base whose behavior is continually updated and reconfigured by its users.

Description of the problem:

To Harmonize: To adjust in fit proportions; to cause to agree; to show the agreement of; to reconcile the apparent contradiction of
. (From Webster's Revised Unabridged Dictionary 1913)

Moving toward Automation: The Harmonized approach

An alternate model to that shown in Figure 2 is demonstrated in Figure 3. In this model, the focus of business automation is the users of the information system. It reflects an overall process that is governed by the mission of the enterprise, the need to improve information management, and an Information Model, which reflects a definition of the information requirements of the user community.

The decision process for each phase of automation is a step within this overall process and can benefit from the lessons learned form previous phases. Technology considerations, which are by nature variable, are also included in the process of implementing the infrastructure but they do not govern the process of automation.
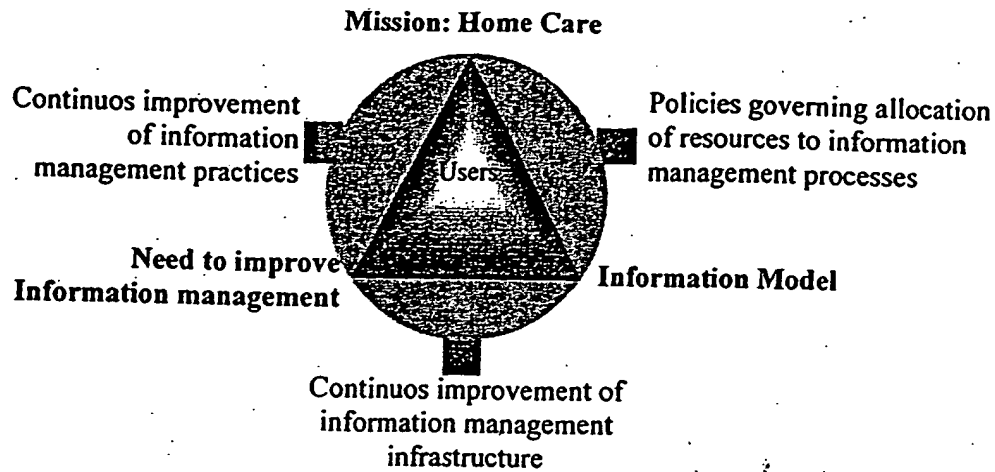
Mission: Home Care

Continuos improvement of information management practices

Users

Policies governing allocation of resources to information management processes

Need to improve Information management

Information Model

Continuos improvement of information management infrastructure

Figure 3. Decision support model based on business considerations

Using this model, the organization defines its automation strategy the basis of the Information Model rather than technology. Tactical considerations balance the need for continuous improvement against the limitations of the available technologies:

1. The overall objective is to improve the effectiveness of the organization toward performance of its mission which is to provide care for patients at their home..

2. The organization's mission and the overall need to improve its ability to manage its information base influence requirements for each phase of automation.

3. The specific goals of each phase of automation are determined by balancing the need to improve information management practices and practical improvements to the organization's view of its information needs. These needs include changes to users' practices as well as affordability of the underlying technology.

4. Policies that govern allocation of resources to each phase of automation are influenced by the organization's model of its information as well as its mission (home care).

The primary function of the invention is creation and continuous improvement of an information system by the individual or organization that deploys a computer system to serve an enterprise. The Information Model:

- Identifies the individual roles within the organization

- Identifies location and characteristics of sources and destination of the information elements that are used or created by each role (the relationship of each role to the data)

- Specifies the rules of translation between elements of data, from the perspective of each individual role.

- Specifies actions that are taken by each role relative to the information (the relationship of roles to one another).

The core components of the invention that support these features are:

1. A persistent record that describes above information for each role: **Configuration**

2. A software program that uses this information to perform the specified actions for each role: **Engine**

3. A formalism that provides for translating the needs of the organization into configuration records for each instance of the Engine: **Process.**

7

The invention is a method and system embodied in a datasource harmonizer that facilitates implementation and deployment of adaptable collaborative enterprise-wide information management system.

It includes a software component that is a program that can operate independent of other programs within a computer system. This program operates as a system service which means that it may be directly exposed to the user of the computer or it may receive its instructions from other application programs that are either running on the computer in question or have the ability to communicate with this computer. (industry term: Engine).

This Engine is implemented as a group of independent, homomorphic components with specific structure and interfacing rules (industry term: Objects). Current embodiment of this basic invention is exemplified in its test implementation within the framework of a Component Object Model and it is itself an Object, that is to say multiple copies of it may be present within the system. {Note: A Component Object Model (COM) object is software industry's analog of an integrated semiconductor circuit. It is an area within the computer's memory that has a specific program and a starting point known to a service handler. A service handler is analogous to a printed circuit board and installing a COM object in the system is analogous to placing the chip on the printed circuit board.}

Using this Engine in various workstations and servers within an enterprise makes it possible for the organization to create an active model of an organization's information use and management practices (an information model). "Active model" implies that this model is defined and modified as needed and in real time to reflect the organization's structure and is an integral part of the organization's daily operation. "Real time" means that in order to re define the responsibilities of one component, other components within the system do not have to stop doing their work. The primary benefit is the ability of the organization that uses this Engine to create an information management system, substantially independent of its specific hardware and software components. Specific claims for the patent that covers this invention will address the elements of the software system as well as the information model and elements of an information architecture that, once adopted by the organization, makes the overall information system a living and changing component of the enterprise.

Advantages of the Invention over the Prior approaches

The datasource harmonizer is a tool for tailoring enterprise applications programs to satisfy the users' requirements it:

- Serves as the agent of the user (not as the glue between applications)
- Satisfies the user's information needs within the enterprise rather than forcing the user to deal with the information that are created by generic applications.
- Defines the enterprise's Information Management practices as an aggregation of the user's information needs.

The datasource harmonizer is a modular software system that supports workflow management, data mapping and application linking. The core component of the datasource harmonizer is an ActiveX® component that is installed in user workstations, laptops or handheld computers throughout an enterprise as needed. The product also includes a utility program for the information manager that is installed in each system or database administrator's workstation. This administrator utility provides the mechanism for the creation and modification of the harmonization rules used by the datasource harmonizer.

Most healthcare organizations have developed processes that are specific to their unique needs using generic business software such as Microsoft® Office as the primary data entry and reporting application. The inventors have implemented the catalogue of business and clinical applications that use Microsoft® Office as a front end and connect to various databases within the enterprise via the datasource harmonizer. These solution packages include customizable forms, templates, add-ins, and application-specific macros and utilities. Each Line-of-Business Solution is marketed to address a specific business or clinical application.

The datasource harmonizer according to the invention easily and economically transforms any single computer, or group of computers, into a scalable information management system. As such, the entire market of more than 2 million computers in healthcare represents the current demand for the problem resolutions of the datasource harmonizer. The inventors' objective is to position its products as the low cost office automation solutions to connectivity used by all facets of the industry.

The datasource harmonizer is designed to address connectivity problems associated with administrative and clinical processes in a distributed computing environment. These problems are currently being addressed on a case by case basis by applications vendors, systems integrators and consultants via complex programming and integration mapping solutions. The datasource harmonizer simplifies the implementation and extends the application of these solutions thereby creating an expanded market opportunity for each of these types of companies. The datasource harmonizer (dsH), is a unique software tool that enables standard desktop office utilities such as Microsoft® Word to manage clinical and administrative workflow and seamlessly map data between multiple healthcare applications and databases. In one embodiment, the engine of the dsH is packaged to work seamlessly with the Microsoft® Office suite of desktop applications. Word processors, spreadsheet programs and personal information management applications (such as e-mail or contact managers) can use the dsH engine to review and input information that up to now has been exclusively available through specialized and complicated applications. Using dsH, business managers can customize the access to and use of information how, when and where they want it.

Most organizations use office automation utilities such as word processors, desktop database managers and spreadsheet programs. Using the dsH technology, the customer can connect these utilities to enterprise databases, legacy applications or simply connect them together to create an enterprise solution.

In contrast, large applications are designed for specific functions within the enterprise but fail to accommodate administrative and clinical processes. dsH provides a simple method for converting a word processor into a powerful report generator or for converting a mail utility into a schedule management tool.

In the last few years it has become common practice for organizations to select vendors based on support of industry standards, primarily in the area of interoperability. The dsH technology leverages this movement and offers an attractive alternative whether the organization is choosing a specific solution or evaluating an enterprise-wide approach to information management.

DETAILED DESCRIPTION OF THE INVENTION

Information Model

To be of value in the decision process, an information system must be easy to implement and adaptable to emerging technologies and changing business needs. Its successful implementation incorporates knowledge of the roles of the individuals and workflow within the enterprise. Additionally, it is necessary to implement such an information system incrementally, allowing for the fact that at all times different parts of the enterprise will be at different levels of operational and technological maturity.

The four basic elements of the invention reflect these concepts in such a way as to facilitate a formal process for creation and maintenance of an Information Model. These elements are Roles, Data Sources, Relations, and Workflow.
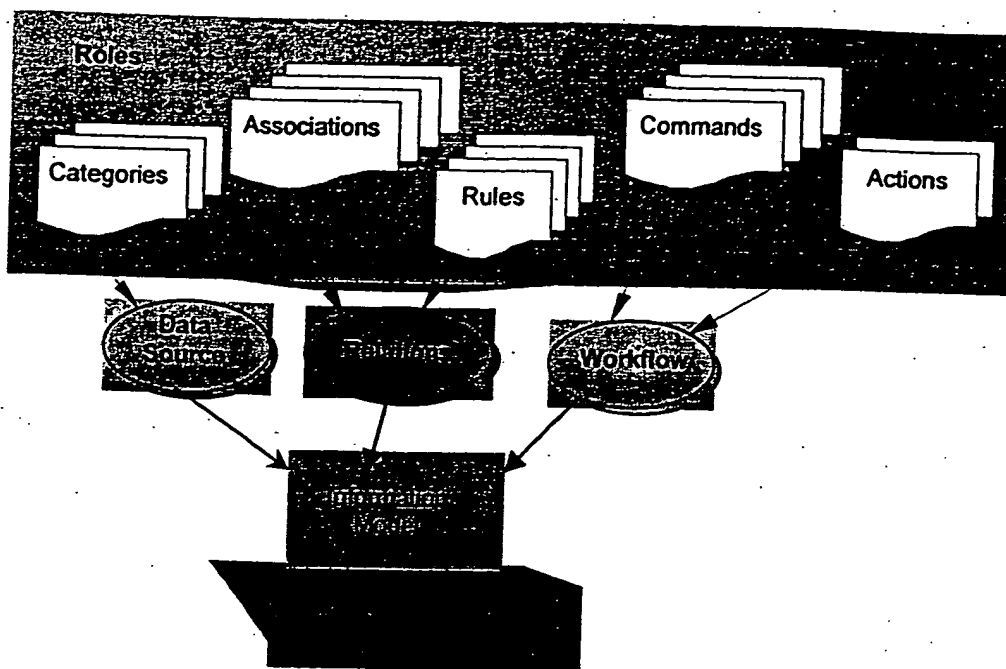
Figure 4 Basic dsH Concepts

The invention provides for definition and utilization of these elements, to the extent that is necessary to create and maintain an Information Model. The focus of the invention is on accuracy of this model and two key qualities of the resulting information system i.e. serviceability and potential for change. Other key qualities of the information system (performance, security, and availability) will reflect the behavior of the implementation platforms.

Roles

The invention provides a mechanism to define a role by specifying the activities that an individual in the organization will require of the information system, in the course of fulfilling an assignment. For example the following statement is a part of the definition of a role:
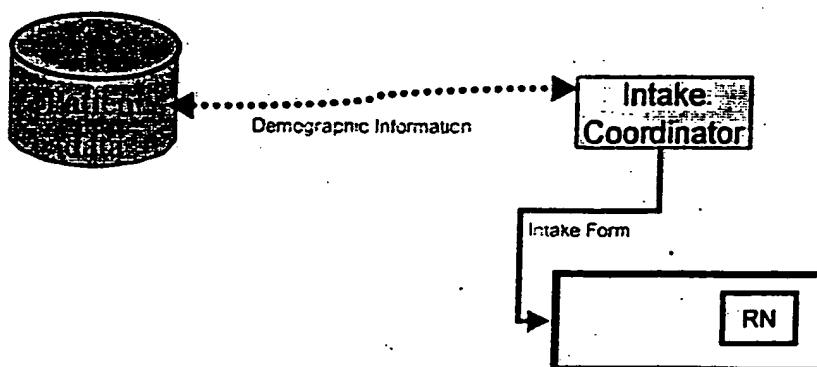

Figure 5 Defining the Intake Coordinator Role

"To OPEN a Case, the Intake Coordinator copies the patient demographic information from patient database into the Intake Form and sends this form to the RN." This statement identifies the following elements of the Information Model:

Data sources	Patient Database and Intake Form
Reference model	the collection of the data fields that describe Patient Demographic information and their relationship to the fields in the Intake Form
Commands	SELECT information defined by patient demographic and INSERT this information into the appropriate fields in intake form
Action	OPEN a Case.

Table 1 Elements of the Intake Coordinator Role

By defining a role in this way, the external sources (to the Information system) where information is created or used are identified. In abstract, a role may be described as "external interfaces of the information system". Figure 5 shows a practical view of a role, as a particular aggregation of the internal elements of the information system.

Data Sources

Data source is an abstract concept that encapsulates the origin and destinations of the information. We have used the generic term "data source", in recognition of Microsoft's ODBC and OLEDB initiatives, which are significant contributors to the computer industry's ability to share information between disparate systems. Our use of this term, however, is broader than Microsoft's. We look at data sources as elements of the entire enterprise rather than a component of the computing environment within the enterprise.

A data source, as a fundamental element of an Information Model refers to any component of the overall organization where information is stored and is made available according to a well-defined set of rules.

Practically, for the purposes of the invention, a data source is defined as any data storage mechanism within the information system for which a formal interface definition is available. This includes databases and application specific file formats and other Engines.

Categories

A reference to a data source within a role is identified as a category. Categories describe the data sources that are used (identification of the logical reference), how to get to them (access parameters), and the way the information within a category is to be treated. For example a category may be a database located at a specific location on a network where the actual data will be accessed. Another category may be a reference schema that describes the way data is organized i.e. *facade*: a view of data into another data source or a report that should be printed or mailed

Relations

Relations are formal, abstract attributes of an Information Model that characterize the relationship between the elements of data sources. The prototype implementation of the Invention included two classes of Relations, namely associations and rules.

Consider the statement:

Patient Name on HCFA 485 form includes Last name, First name and Middle initial from the patient record. HCFA requires that Patient Name is specified as Last name, First name middle initial

This statement describes the Relation between the element *Patient Name* in data source *HCFA 485 form* and the elements *First name, Middle initial,* and *Last name* in the data source *patient record.*

11

Associations

An association is a declaration that a relationship exists between two or more elements of different data sources. In the above example the following associations exist:

HCFA 485 form   *Patient record*
Patient Name   *First name*
Patient Name   *Middle initial*
Patient Name   *Last name*

Associations do not imply sequence or cardinality.

Rules

Rules are the mechanisms that embody an Association. In the above example the Rule relating to the association is:

Name <= Last_name<,> First_name <sp> Middle_initial).

In general, Rules are attributes of Relations and independent of Associations. At the simplest level, this independence my be perceived as a means to re-use a definition of a Rule in the context of multiple Association. Although this is an appropriate example of the relationship between a Rules and Associations it does not completely capture the scope of this concept. For example the Invention allows for implementation of a time-variant, parametrically defined Rule that creates independent events based on a change in values of elements within multiple data sources.

Workflow

The Workflow Management Coalition (WFMC) defines Workflow as: "The computerized facilitation or automation of a business process, in whole or part". In the context of the Invention, Workflow refers to those attributes of an Information Model that are concerned with movement of information between different data sources. To a great extent, Workflow is the element of the Invention that creates a unique instance of the Harmonizer Engine, specific to the individual User. The prototype implementation of the Invention included two classes of Workflow elements, namely Commands and Actions. As a general rule, Workflow elements tend to serve as a "verb" in a sentence where Categories serve as nouns.

Commands

Commands are the lowest level operators for moving data from one data source to another. The most common examples of commands in an Information System are *Select* and *Insert*. As a rule, Commands require parameters that qualify their targets. In the example of section 4.1.3 a Command may be:

select *First_name, Middle_initial_Last_name* from patient_record where *Patient_ID* = this_patient

insert patient_name into HCFA 485 form

Actions

Actions are aggregations of commands that are visible to the run time interfaces of the Engine. In general Actions are used to activate the portion of the Information Engine that is specific to an individual user. Actions are generally high level events and include general parameters that define the parameters of the data sources and the context within which information is modified:

Nurse_suzie **Admit** this_patient

Prototype implementation: The engine

The Engine is an Active component, such as a program, that moves information between various data sources within a network. A "data source" can be any collection of data for which a provider can be created (such as a tab separated files, spreadsheets, named fields in a form, or a table in a relational database). Engines are capable of communicating information in form of data and status i.e. an Engine can see data sources through other Engines and can send and receive events.

An Engine may be used as a single utility that simplifies access to various data sources for a single user or device. An example of this kind of application may be a report that is created in a document that uses information from various databases, PACS systems and Hospital Information Management systems. Another example may be a protocol converter for a device within a health information management system. The Engine may also be used as a member of a collection of Engines and facilitate workflow as well as information access. An example of such an application is a Long-Term Care environment where various healthcare professionals communicate with each other through sending and receiving pre-specified forms. Each Engine uses a configuration model file that specifies its associated datasources, trigger events, and mapping rules.

Using the Engine a user of a networked workstation can input, export and review information form different databases or documents. The user interface may be desktop applications, Word processors, spreadsheet programs and personal information management applications (such as e-mail or contact managers) as well as many commercially available special purpose applications.

Implementation

Overview

The dsH engine provides the ability to associate or map together various data sources, apply rules to the to the associations and move information between data sources programmatically. The engine exposes this functionality through a set of COM interfaces designed to allow design time configuration, a single universal data source interface and a simple interface to execute the pre-configured information movements. The interfaces are designed to address the needs of several types of users.

Interface	Medical Professional	IT Programmer	Partner Developer	Inventors' Developer
Execute Interface	X	X	X	X
OLEDB Provider Interface		X	X	X
Configuration Interface		X	X	X
OLEDB Consumer Interface			X	X
Persistent Layer Interface				X

OLEDB Provider Interface

The OLEDB Provider Interface allows the user to connect to the dsH Engine as a data source. The schema exposed is determined by the configuration of the harmonization model. This interface has two uses in the overall design. First it allows the end user to access any dsH data source through standard database tools found in MSOffice™ or development environments. The second use is to dsH Engine to treat another dsH Engine as a data source.

OLEDB Consumer Interface

The OLEDB Consumer Interface provides for the connectivity to all external data sources and is the only connection method supported. The dsH Engine can connect to any OLEDB/ODBC provider.

Configuration Interface

The Configuration Interface (a.k.a. Utility interface) exposes the programming model that allows the creation of the data source harmonization model and the definition of its execution behavior. This interface can be exposed fully for the Partner Level user to develop products that make use of the dsH Engine. It is also exposed in a limited way through wizard to allow the IT Programmer Level and Medical Professional Level users to update and modify their harmonization models.

Execute Interface

The purpose of the Execute Interface is to provide a simple interface from which to cause Actions to be started within the dsH Engine. This interface has one method which starts and Action and one event which notifies the consumers code that the action has been executed.

Persistent Layer Interface

The storage of the harmonization model used by the dsH Engine is stored in a file. The Persistent Layer Interface allows for the support of different methods of storage.

Information Modeling

Information Modeling is the process of maintaining a formal description of how information within an enterprise is created, who creates it, and in what forms it is used. Ultimately an Information Model is a tool that will help with the process of continuously improving the operation of the organization. As a tool of process improvement, the Invention offers four key concepts that are somewhat independent and at the same time collectively provide a complete definition. These key concepts are identified in this section:

- Roles: The complete definition of the creators and users of information

- Categories: Data sources, schema presentation formats of data containers.

- Messages: Information transport media

- Actions: Events that govern transmission of messages between roles

These concepts are "abstractions" in the sense that each of them is a convenient aggregation of ideas and practices that satisfy the needs of a particular view within the enterprise.

Roles

The most basic concept in the Invention is the concept of a Role. A Role is a description of a collection of assignments, usually associated with an individual or group with a specific set of credentials. Table 2 is an overview of the relationship between a role and the flow of information.

Abstraction	Described in terms of...
Role	Qualifications of the individuals e.g. Registered Nurse, Certified Records Administrator
Assignments	Objectives described qualitatively e.g. Admit a patient, Maintain patient records.
Tasks	Measurable goals e.g. complete intake form, perform initial case assessment.

Table 2

To satisfy the needs of creating and maintaining and Information Model, identification of a given role is sufficient. In fact deeper abstractions will create an obstacle to exploring alternative approaches within the overall objective of continuos improvement. This is in contrast, for example, to the process of application development where it is necessary to categorize the tools that are needed in performing each task.

The fact that the end users of the Invention product are practitioners within a regulated industry makes the concept of a "Role" a key concept in Information Modeling. Within this industry, the qualification of an individual practitioner, more than the business considerations, determines the assignments that they can accept and the information that they need in order to meet the objectives of those assignments.

Categories

Within the organization, however, different roles can collaborate because they share *perspectives* on information. A *perspective* is an agreement on how to view information. A category is an abstraction that identifies a "view" of the information that may be useful to a group of perspectives.
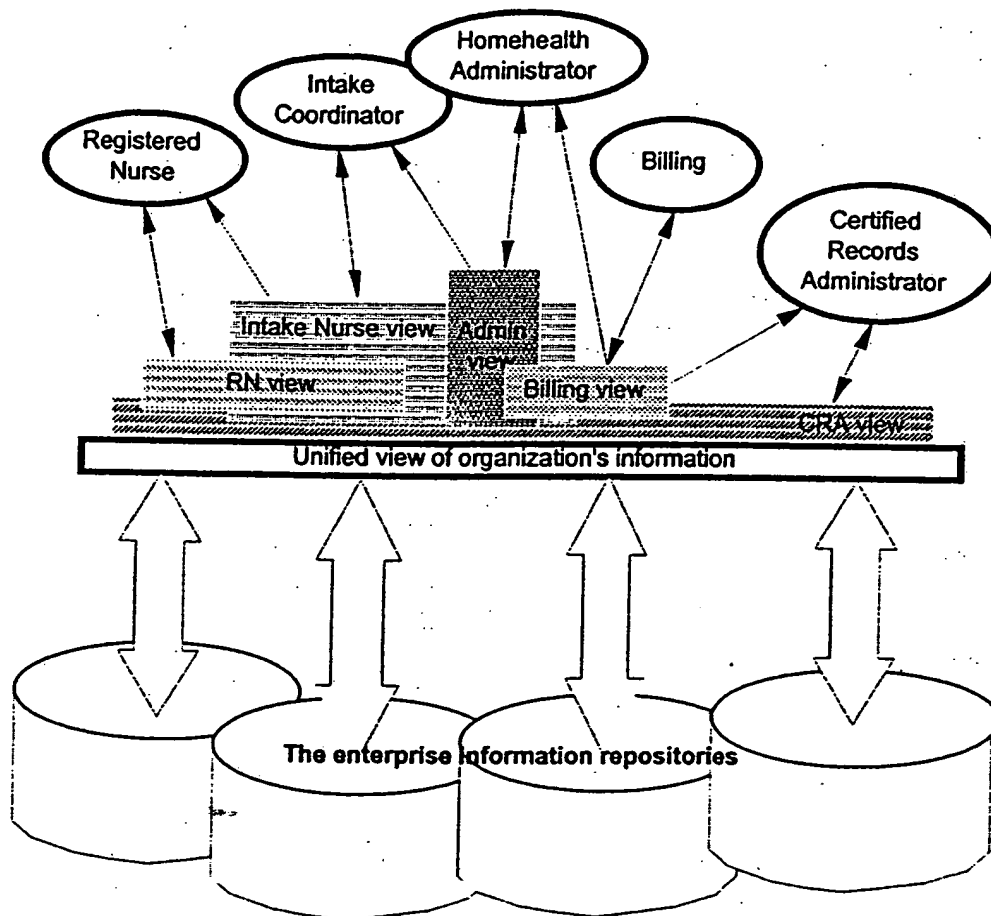

Figure 6 Perspectives and Views of Information

Using this framework, the following concepts are of interest in defining categories.

- Data repositories

- Data repositories as viewed through roles

- Reference categories

- Facades

The first two concepts are concerned with the sources of information whereas the last two deal with creation and maintenance of the Information Model itself. A good general rule is design of an Information Model is that reference categories are inventions that facilitate dealing with data repositories and facades are inventions that facilitate dealing with roles.

Data Repositories

For the purposes of Information Modeling any persistent medium that may contain information is a data repository. For practical purposes the information flow in and out of a given data repository is subject to its specific characteristic. For example in the context of an automated document management system, a hand written document that will be stored as apart of patient's medical record is a valid data repository with a somewhat complex interface rules. For the purposes of Information Models related to the use of the Invention, machine accessible data repositories are the primary consideration.

Repositories viewed through roles

For all purposes, roles appear as sources of information to other roles. In practice, however, it is the view of information from within a role that is of interest to other roles.
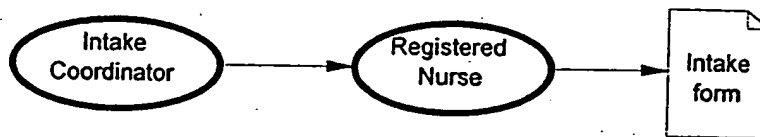

Figure 7 Repositories viewed through Roles

For example in figure 7 the *Intake Coordinator* can see *Intake form* through the role *Registered Nurse*. Here, Registered Nurse is providing a data source to the Intake Coordinator. In practice, this concept is best implemented through the use of Facades, which makes it possible to manage access to data sources.

Reference Categories

Two significant problems in Information Modeling are initial creation and maintenance of the model. The reference category is a useful invention to help with this problem. Typically a group of roles within an organization contribute to accomplishing a particular objective. For example the Intake Coordinator opens a case and assigns it to a registered nurse that will then make an assessment and creates the care program. During the period that the case is open various professionals will access and input information related to it.
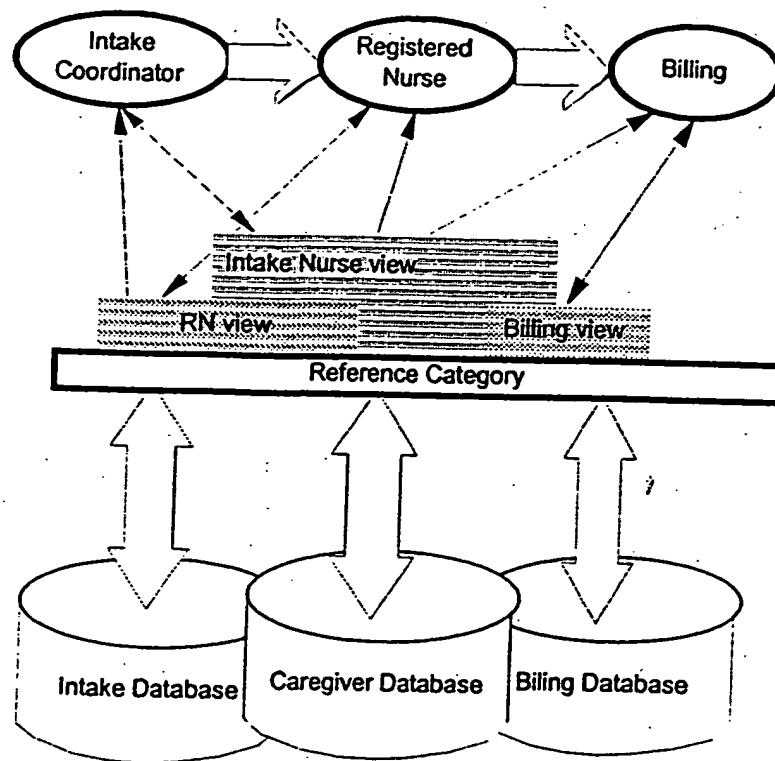
Figure 8 Reference Category

The Information Model is based on information that is used or created by each role within the organization. The Reference Category provides a mechanism to aggregate the information that is used by a group of professionals in the course of completing an assignment under a single category. This mechanism allows a particular data repository to be redefined without effecting the rest of the Information Model.
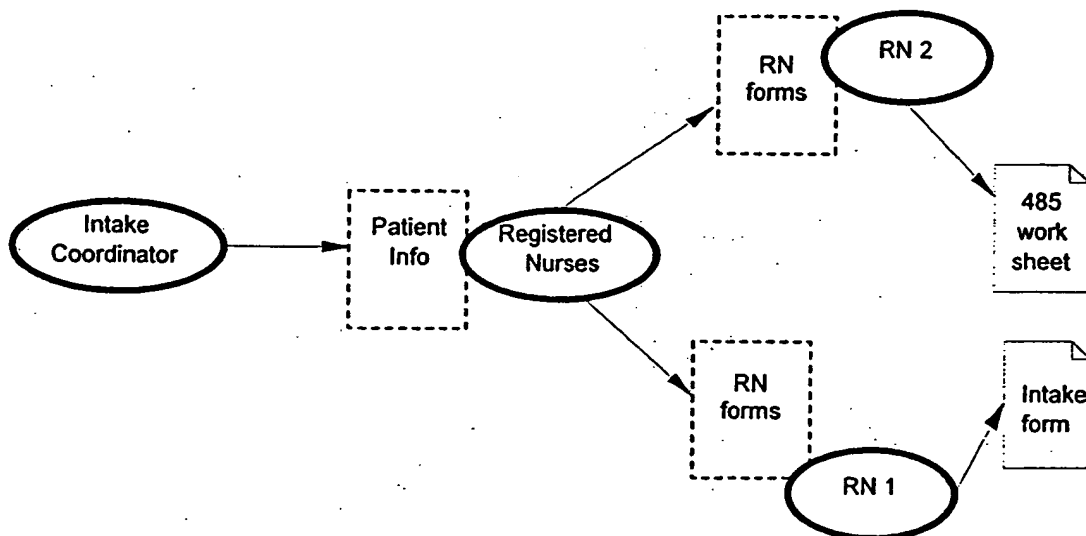
Facades

Figure 9 Facades

In figure 9 The category "patient info" is a façade, that represents information content of forms used by Registered Nurses, from the point of view of the Intake Coordinator. The category "RN forms" is the information content of forms 485 worksheet or Intake forms that are common to all registered nurses.

Using Facades makes it possible for RN1 role to begin using 485 worksheet without affecting other roles in the Information Model.

Of course, the above description of the invention can be implemented in a number of other ways and embodiments in addition to those discussed above including method and systems that are integrated into a single device/method or in modules. The following is brief description of one embodiment of implementation of the invention.

The datasourceHARMONIZER (dsH) system is based on distributed application architecture and includes modular components that support workflow management, data mapping, and application linking. These components implement an information management system that is effective in single workstations, centralized databases, and clusters of workstations using client-server architecture.

A core component of the dsH system is the dsH engine, an ActiveX® component that is installed in user workstations, laptops or handheld computers.

While systems administrators managing enterprise-wide applications use these components for tactical data access solutions, end users of this information management system only see familiar generic office applications like Microsoft® Word or Outlook®.

The following features and elements are included in the dsH system:

dsH Client - The dsH Client is an office automation tool that enables standard office utilities to be used in conjunction with a business's processes to transform any workstation or group of workstations into a scalable information management system.

dsH Administrator Utility - The dsH Administrator Utility is a program for the information manager that is installed in each system or database administrator's workstation. This utility provides the mechanism for the creation and modification of the harmonization rules used by the dsH™ engine.

Line-of-Business Solutions - These pre-built packages include customized (and customizable) forms, add-ins, and application-specific macros and/or utilities for use with generic business software such as Microsoft® Office. These turn-key packages include, for example:

- An ADT (admit/discharge/transfer) package that includes an interface for entering or updating patient information and history.

- A caregiver package that includes scheduling and reporting of delivering services.

- A records administrator package that allows review of census data.

dsH Engine - dsH engine is a core component of the dsH system architecture that facilitates access to multiple databases from standard desktop utilities such as Microsoft® Office products. It will be introduced with support for the following industry standards:

• Database interfaces to include HL7, DICOM, ASTM, and specific Virtual Medical Devices within the IEEE 1073 family of standards as well as ODBC and OLE DB protocols.

• Microsoft® Windows® (95, 98, NT and 2000) platforms.

• Desktop utilities include Microsoft® Word, Access, Excel and Outlook®.

System Designer Workbench (SDW) - SDW provides buyers with a simple tool that enables them to describe their requirements and the SDW software selects the appropriate components. SDW™ components include a database that maintains the customer's configuration and a prototyping tool used to explore alternative configurations. This capability supports the following features: Component selection and purchasing; Remote configuration management; and On-line IT management

Significantly, the invention's system architecture strongly supports extensions to other platforms, database interfaces and desktop applications and utilities.

Another Preferred Implementation of the dsH Technology

In another preferred embodiment, the inventors have implemented the dsH component technology based on Microsoft®'s Component Object Model (COM) initiative. Using COM components, the implementation leverages Microsoft®'s OLE DB for uniform data access, making data universally available to the entire Microsoft® Office product family.
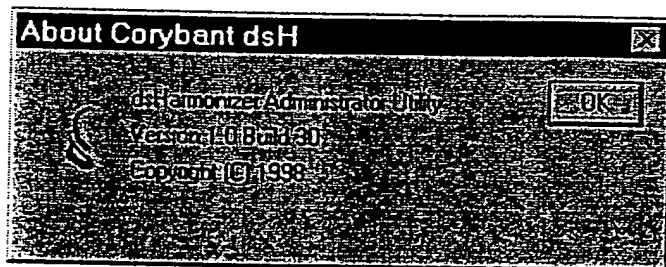
The data interface layer is generalized so that the data source may be an object database, a flat file, a Relational Database Management System (RDBMS), or a standard interchange protocol (HL7, DICOM, ASTM, etc.). Additional application-specific interchange protocols can be dynamically added.
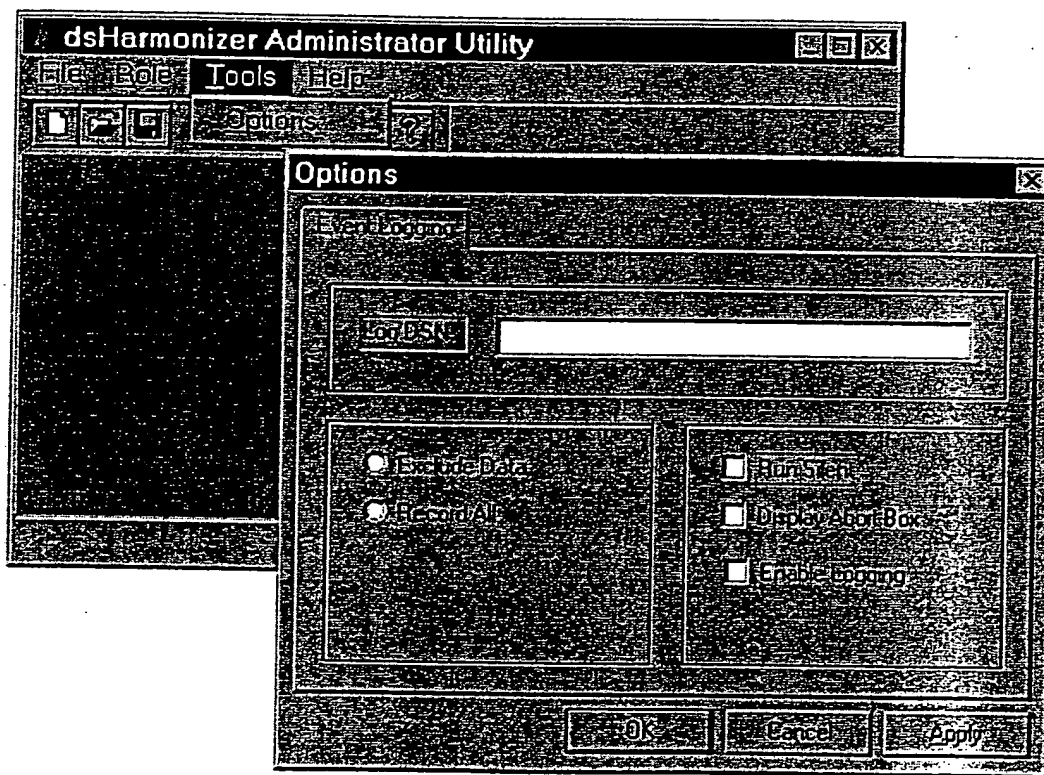
The dsH also provides data dictionaries with built-in drivers for healthcare standards. These data dictionaries enable the components to locate the data classes (both physically and structurally) and extract the information content by applying knowledge-based rules to that data.

Additionally, the dsH provides an administrator utility to control access rights, incorporate changes, map data, and customize interfaces for each provider or group of providers. Healthcare providers then work with just the familiar user interfaces of the Microsoft® Office applications.
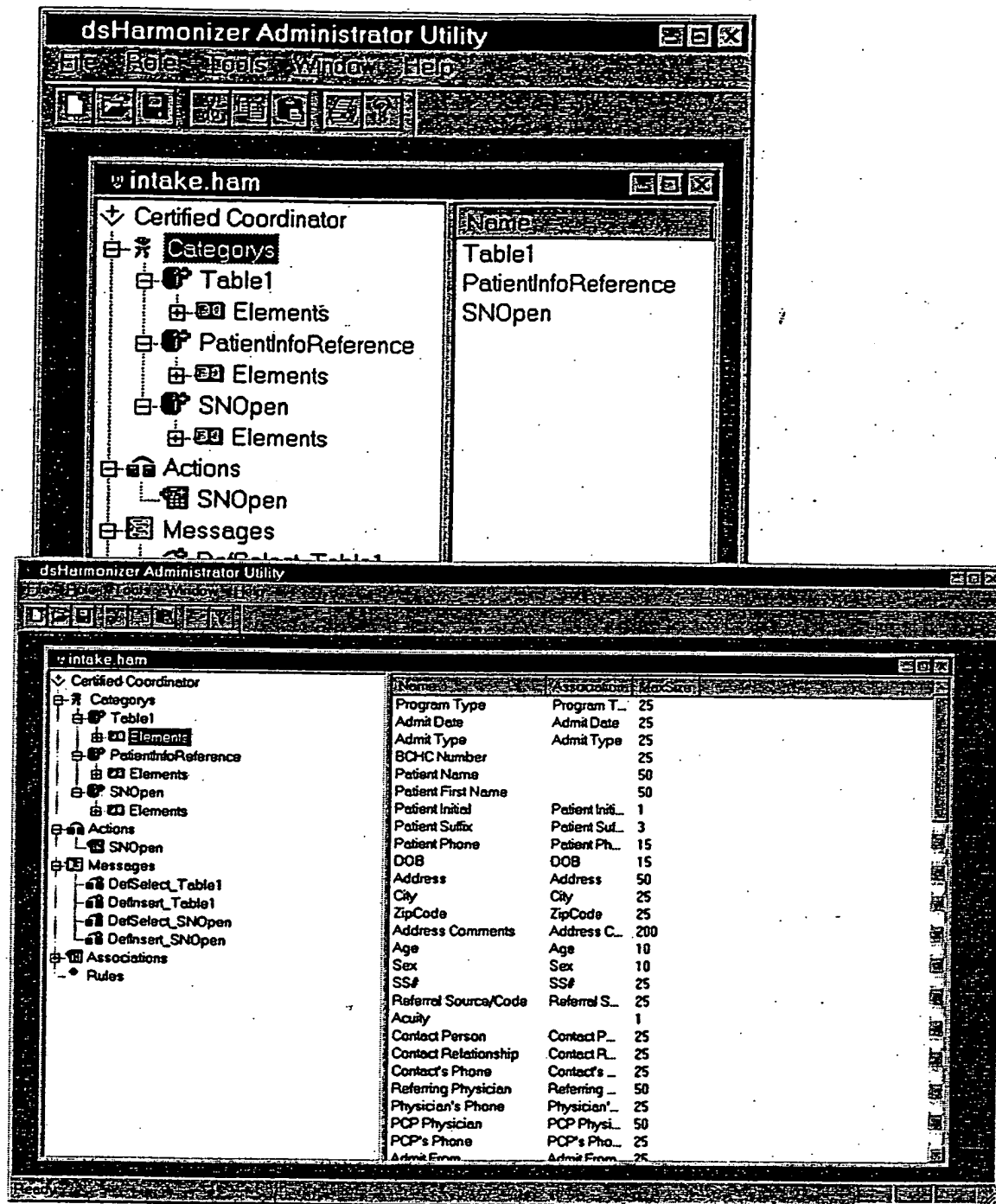
The inventors recognize that numerous programming languages, software suites and applications, and computer hardware can be used to create and operate the invention. In one mode of operation, the inventors utilized the following tools as an environment for building the invention: WinNT 4.0 with SP 4; Visual C++ 6.0 with SP3/ATL 3.0 and MFC 4.2; Visual Basic 6.0 SP3; MDAC 2.1 SP1; and Installshield For Visual C++ 6.0. The inventors also have tested the invention and have proven the methods and systems to effectively and efficiently operate in the test environments of: Win98 with Install MDAC 2.1 SP1 and Win95 with latest SP, Install DCOM, and Install MDAC 2.1 SP1.

The following are screen shots of the utility program are provided for further description and disclosure of screens visible to a user in one embodiment of the invention that was successfully tested.

dsHarmonizer Administrator Utility 🗔🗖☒
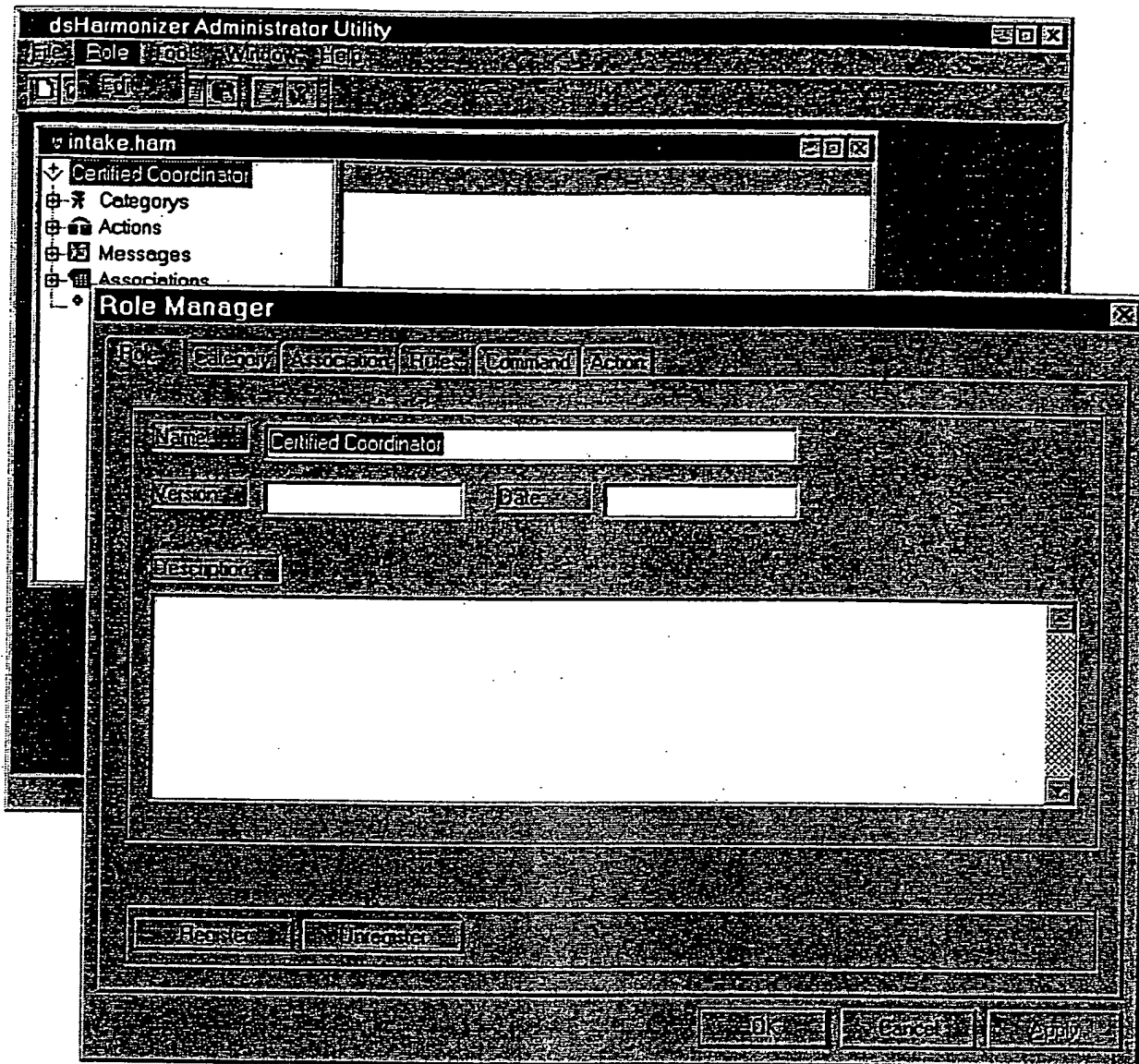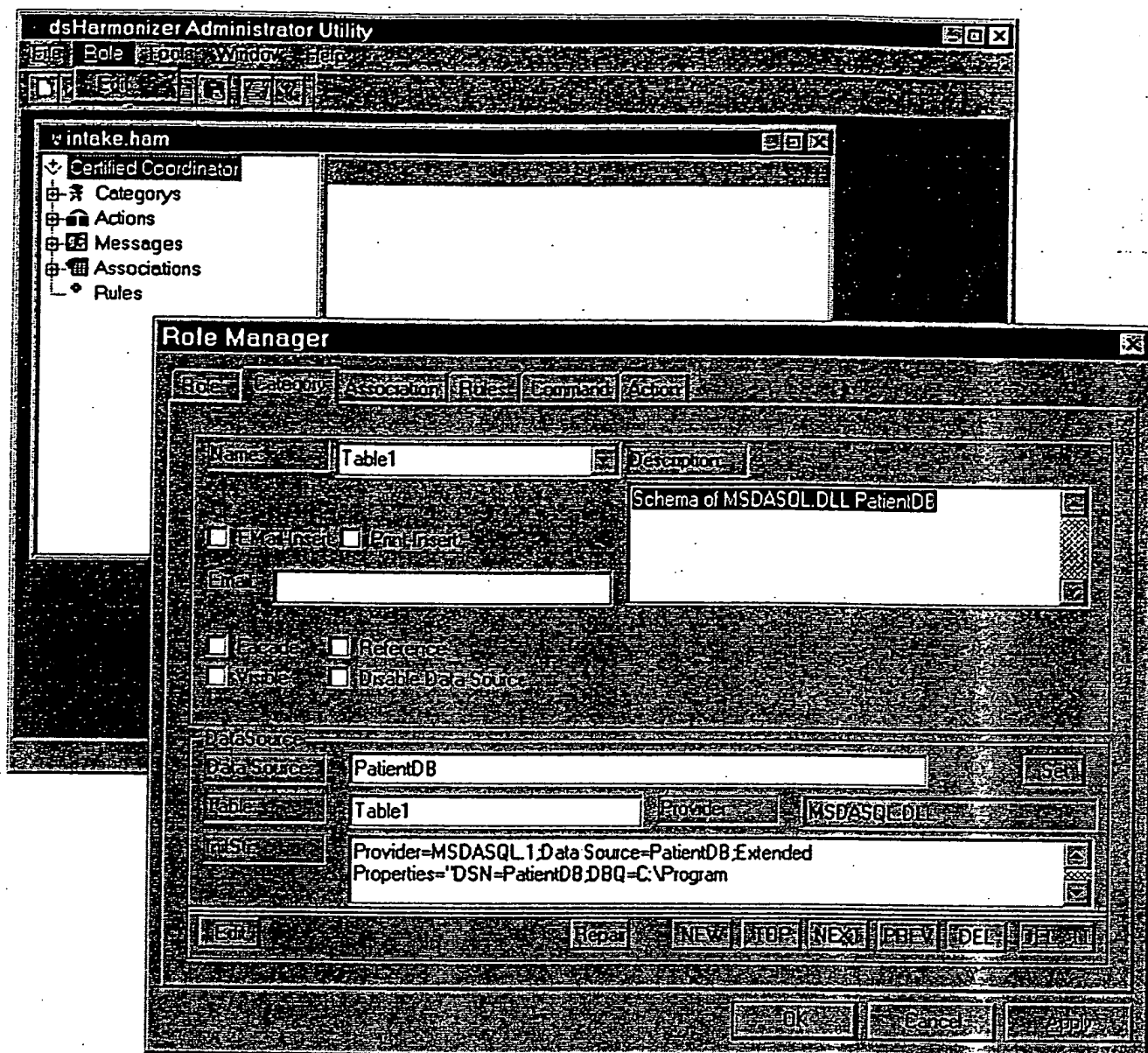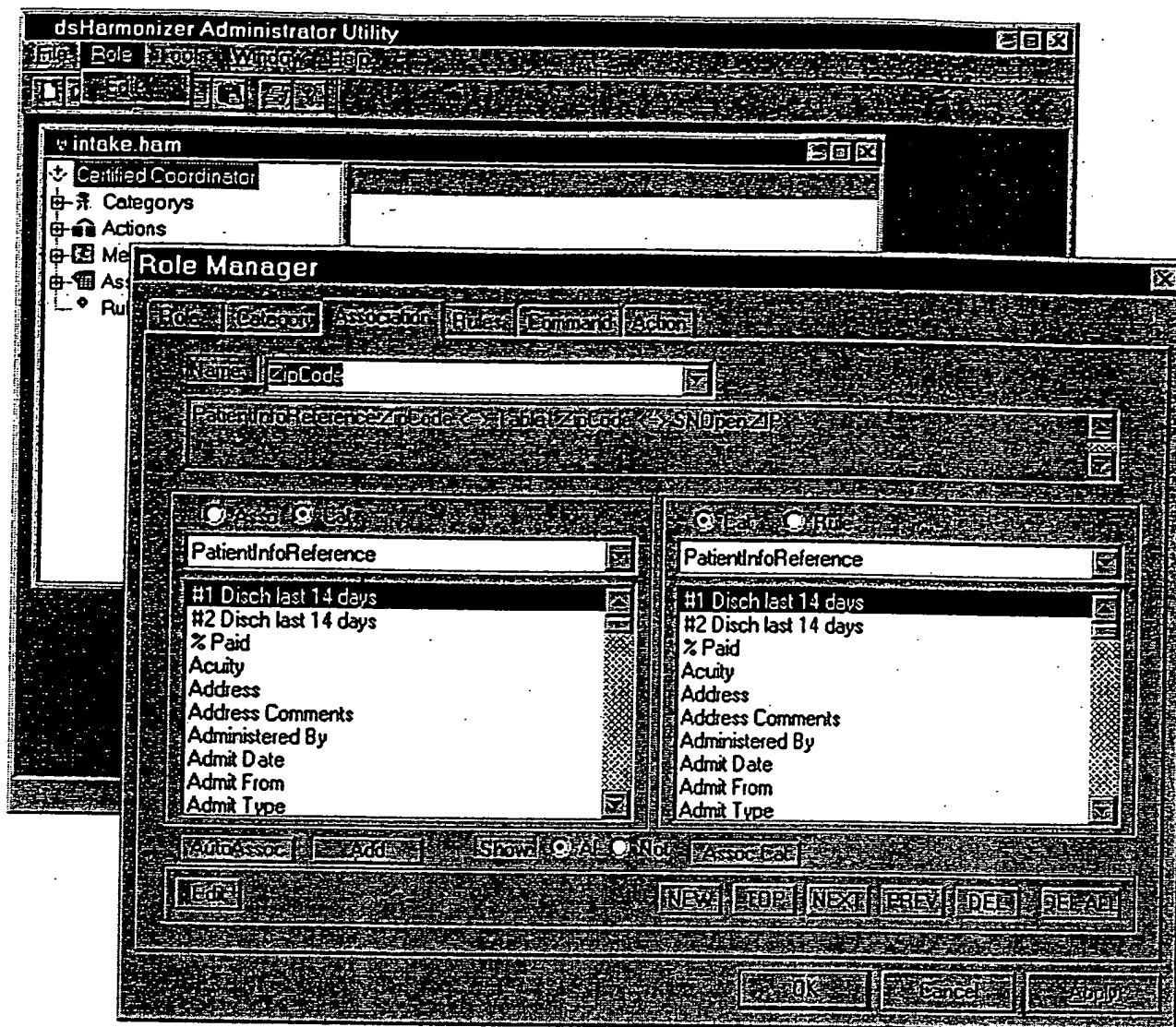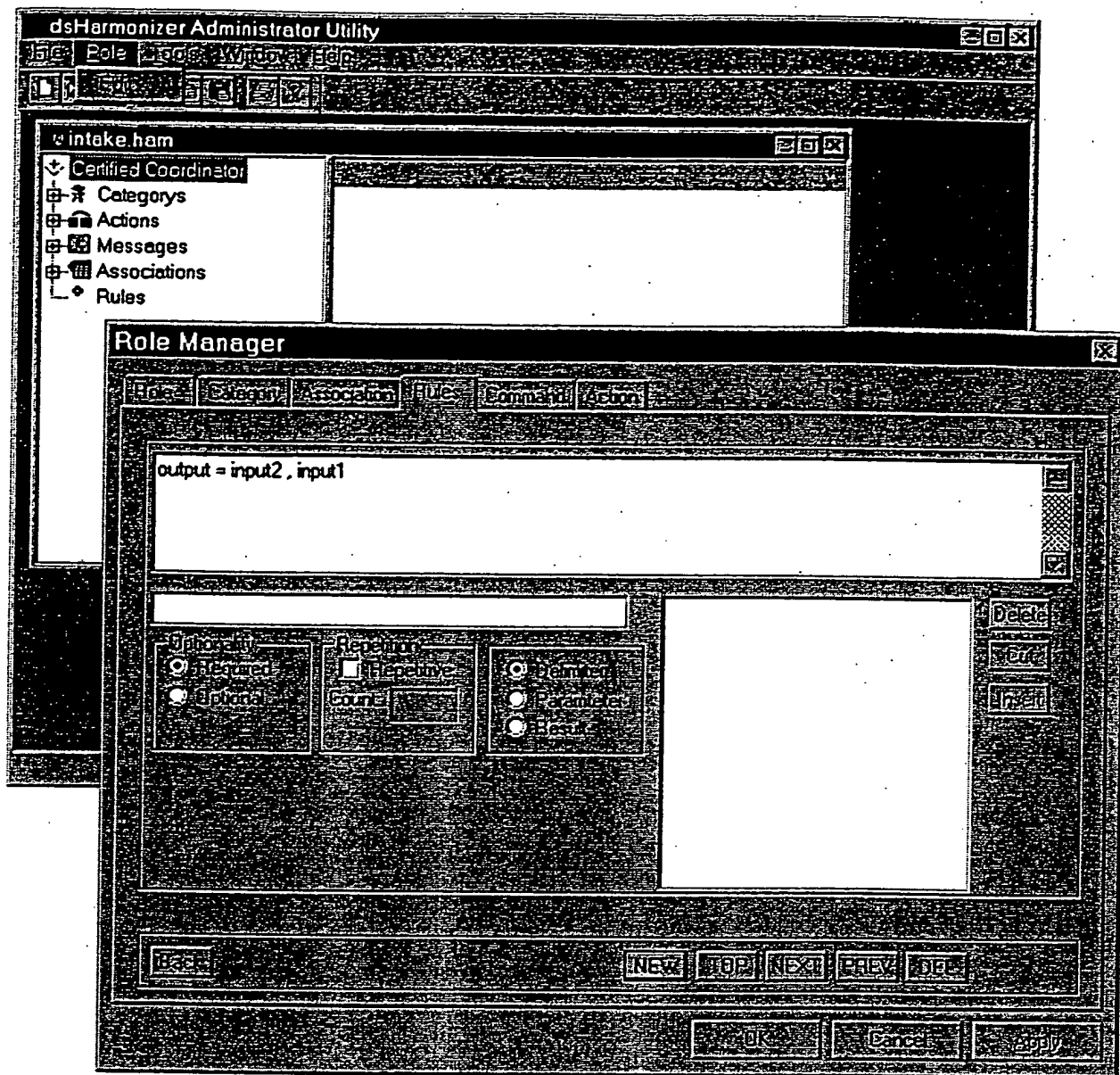
File Role Tools Help

Options ?

Options ☒

Event Logging

Log DSN:

⦿ Echo to Data   ☐ Run Silent
⦿ Record All   ☐ Display Abort Box
                        ☐ Enable Logging

OK   Cancel   Apply

20

Navigation view:

dsHarmonizer Administrator Utility

File Role Tools Window Help

intake.ham

Certified Coordinator
- Categorys
  - Table1
    - Elements
  - PatientInfoReference
    - Elements
  - SNOpen
    - Elements
- Actions
  - SNOpen
- Messages

Name
Table1
PatientInfoReference
SNOpen

dsHarmonizer Administrator Utility

File Role Tools Window Help

intake.ham

Certified Coordinator
- Categorys
  - Table1
    - Elements
  - PatientInfoReference
    - Elements
  - SNOpen
    - Elements
- Actions
  - SNOpen
- Messages
  - DefSelect_Table1
  - DefInsert_Table1
  - DefSelect_SNOpen
  - DefInsert_SNOpen
- Associations
- Rules

Name	Association	MaxSize
Program Type	Program T...	25
Admit Date	Admit Date	25
Admit Type	Admit Type	25
BCHC Number		25
Patient Name		50
Patient First Name		50
Patient Initial	Patient Initi...	1
Patient Suffix	Patient Suf...	3
Patient Phone	Patient Ph...	15
DOB	DOB	15
Address	Address	50
City	City	25
ZipCode	ZipCode	25
Address Comments	Address C...	200
Age	Age	10
Sex	Sex	10
SS#	SS#	25
Referral Source/Code	Referral S...	25
Acuity		1
Contact Person	Contact P...	25
Contact Relationship	Contact R...	25
Contact's Phone	Contact's ...	25
Referring Physician	Referring ...	50
Physician's Phone	Physician'...	25
PCP Physician	PCP Physi...	50
PCP's Phone	PCP's Pho...	25
Admit From	Admit From	25

Ready

Edit Views:

dsHarmonizer Administrator Utility

File Role Tools Windows Help

intake.ham

- Certified Coordinator
- Categorys
- Actions
- Messages
- Associations

Role Manager

Role | Category | Association | Titles | Command | Action

Name: Certified Coordinator

Version: Date:

Description:

Register   Unregister

OK   Cancel   Apply

dsHarmonizer Administrator Utility

File Role Tools Window Help

intake.ham

- Certified Coordinator
  - Categorys
  - Actions
  - Me
  - As
  - Ru

Role Manager

Role | Category | Association | Rules | Command | Action

Name: ZipCode

PatientInfoReference.ZipCode <=> Table.ZipCode <=> SNOpenZip

○ Assoc ● Cat   ○ Cat ○ Rule

PatientInfoReference   PatientInfoReference

#1 Disch last 14 days
#2 Disch last 14 days
% Paid
Acuity
Address
Address Comments
Administered By
Admit Date
Admit From
Admit Type

#1 Disch last 14 days
#2 Disch last 14 days
% Paid
Acuity
Address
Address Comments
Administered By
Admit Date
Admit From
Admit Type

AutoAssoc | Add | Show ● All ○ Not | Assoc Cat

Edit | NEW | TOP | NEXT | PREV | DEL | DEL ALL

OK | Cancel | Apply

24

dsHarmonizer Administrator Utility

File Role Tools Window Help

intake.ham

Certified Coordinator
- Categorys
- Actions
- Messages
- Asso... Associations
- Rule... Rules

Role Manager

Role | Category | Association | Rules | Command | Action

Name: DefSelect_Table1

Status: corrupted

SQL | Procedure | Clear Default | $800 | Qualifier: ?

Command:

AND	SNOpen
WHERE	
INSERT	Acuity
SELECT	Address
	AddressComments
	AdministeredBy
	AdmitDate
	AdmitForm
	AdmitType
	Age
	BCHCNumber

SELECT [Program Type], [Admit Date], [Admit Type], [BCHC Number], [Patient Name], [Patient First Name], [Patient Initial], [Patient Suffix], [Patient Phone], [DOB], [Address], [City], [ZipCode], [Address Comments], [Age], [Sex], [SS#], [Referral Source/Code], [Acuity], [Contact Person], [Contact Relationship], [Contact's Phone], [Referring Physician], [Physician's Phone], [PCP Physician], [PCP's Phone], [Admit From], [Inpatient start date], [Inpatient end date], [Comments], [Primary Diagnosis

Context: 11.00   Test | Retain | NEW | ADD | NEXT | PREV | DEL

OK | Cancel | Apply

dsHarmonizer Administrator Utility

File Role Tools Window Help

intake.ham

- Certified Coordinator
- Categorys
- Actions
- ...

Role Manager

Role | Category | Association | Rules | Command | Action

Name: SNOpen

Commands: ADD Actions:

DefInsert_SNOpen
DefInsert_Table1
DefSelect_SNOpen
DefSelect_Table1

DefSelect_Table1 -> DefInsert_SNOpen

Context: 11.00   Role: m010   Test

New TOP NEXT PREV DEL

OK Cancel Apply

VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(b))—INDEPENDENT INVENTOR	Docket Number (optional) 19382-002

Applicant or Patentee: RONALD M. BARBER, BARBARA J. ROSSNER, and MOHAMMAD S. SALIM

Serial or Patent No.: Not Yet Accorded

Filed or Issued: Not Yet Accorded

Title: DATASOURCE HARMONIZER

As below named inventors, we hereby declare that we qualify as independent inventors as defined in 37 CFR 1.9(c) for the purposes of paying reduced fees to the Patent and Trademark Office described in:

[X] the specification filed herewith title as listed above.

[ ] the application identified above.

[ ] the patent identified above.

I have not assigned, granted, conveyed or licensed and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

[ ] No such person, concern, or organization exists.

[X] Each such person, concern or organization is listed below.

Corybant, Inc.

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the of the application, any patent issuing thereon, or any patent to which this verified statement is directed.
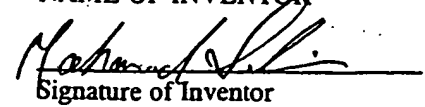
RONALD M. BARBER	BARBARA J. ROSSNER	MOHAMMAD S. SALIM
NAME OF INVENTOR	NAME OF INVENTOR	NAME OF INVENTOR
Signature of Inventor	Signature of Inventor	Signature of Inventor
1/9/00	1-19-2000	1/19/2000
Date	Date	Date

FORM PTO 1595
1-31-92

RECORDATION FORM COVER SHEET
PATENTS ONLY

U.S. DEPARTMENT OF COMMER
Patent and Trademark Off

To the Honorable Commissioner of Patents and Trademarks: Please record the attached original documents or copy thereof.

1. Name of conveying party(ies) Ronald M. Barber, Barbara J. Rossner, and Mohammad S. Salim Additional name(s) of conveying party(ies) attached? [ ] Yes [X] No	2. Names and address of receiving party(ies): Name: Corybant, Inc. Address: 3800 Arapahoe, Suite #205 City, State & Zip: Boulder, CO 80303

3. Nature of conveyance: [X] Assignment [ ] Merger [ ] Security Agreement [ ] Change of Name [ ] Other _____ Execution Date __January 19, 2000__	Additional names & addresses attached? [ ] Yes [X] No

4. Application number(s) or patent number(s):

If this document is being filed together with a new application, the execution date of the application is: __January 19, 2000__

    A.   Patent Application No.(s)   B.   Patent No.(s)

7725!

CHRISMAN, BYNUM & JOHNSON, P.C. 6-73
ATTORNEYS AT LAW
P.O. BOX 4429 PH. 546-1300
BOULDER, CO 80306

BANK ONE, COLORADO, NA
DENVER, CO 80202
23-101-1020

PAY   *FORTY AND 00/100 DOLLARS*

TO THE
ORDER OF

DATE		AMOUNT
01-19-00	77254	*******$40.00

ASSISTANT COMMISSIONER FOR PATENTS
BOX ASSIGNMENT
WASHINGTON DC 20231

⑆077255⑈ ⑆102001017⑆ 106900215011⑈

Stefan I. Dump		1-11-2000
Name of person signing	Signature	Date
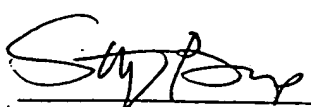
Total number of pages comprising recordation: [3]

RECORDATION FORM COVER SHEET
PATENTS ONLY

U.S. DEPARTMENT OF COMME
Patent and Trademark O

To the Honorable Commissioner of Patents and Trademarks: Please record the attached original documents or copy thereof.

1. Name of conveying party(ies) **Ronald M. Barber, Barbara J. Rossner, and Mohammad S. Salim** Additional name(s) of conveying party(ies) attached? [ ] Yes [X] No	2. Names and address of receiving party(ies): Name: **Corybant, Inc.** Address: **3800 Arapahoe, Suite #205** City, State & Zip: **Boulder, CO 80303**

3.   Nature of conveyance:

    [X] Assignment   [ ] Merger

    [ ] Security Agreement   [ ] Change of Name

    [ ] Other _____

    Execution Date ____ **January 19, 2000**

Additional names & addresses attached? [ ] Yes [X] No

4.   Application number(s) or patent number(s):

If this document is being filed together with a new application, the execution date of the application is: **January 19, 2000**

A.   Patent Application No.(s)   B.   Patent No.(s)

Additional numbers attached? [ ] Yes [X] No

5. Name and address of party to whom correspondence concerning document should be mailed: Name: **Thomas C. Folsom, Esq.** **Chrisman, Bynum & Johnson, P.C.** Street Address: **1900 Fifteenth Street** City, State, Zip: **Boulder, CO 80302**	6. Total number of applications and patents involved: 1
	7. Total fee (37 CFR 3.41) $40.00 [X] Enclosed [ ] Authorized to be charged to deposit account
	8. Deposit account number: 03-1725 (Attached duplicate copy of this page if paying by deposit account)

DO NOT USE THIS SPACE

9.   Statement and signature.
To the best of my knowledge and belief, the foregoing information is true and correct and any attached copy is a true copy of the original document.

Stefan T. Bump
Name of person signing

Signature

1-19-2000
Date

Total number of pages comprising recordation: [3]

ASSIGNMENT

WHEREAS, we, Ronald M. Barber, 2988 South Lakeridge Trail, Boulder, CO 80302; Barbara J. Rossner, 1301 30th Street, #102, Boulder, CO 80303; and Mohammad S. Salim, 2211 Bluebell Ave., Boulder, CO 80302, have invented a certain new and useful DATASOURCE HARMONIZER, for which we are about to make provisional application for Letters Patent of the United States, said provisional patent application being identified as File No. 19382-002, in the files of the Law Offices of CHRISMAN, BYNUM & JOHNSON, 1900 Fifteenth Street, Boulder, CO 80302 and executed concurrently herewith;

WHEREAS, Corybant, Inc., 3800 Arapahoe, Suite #205, Boulder, Colorado 80303, a Colorado corporation, is desirous of acquiring the entire right, title and interest therein and thereto;

NOW, THEREFORE, be it known that for and in consideration of the sum of One Dollar ($1.00) and certain other good and valuable consideration to us in hand paid, the receipt of which is hereby acknowledged, we, the said Ronald M. Barber, Barbara J. Rossner, and Mohammad S. Salim, by these presents do sell, assign and transfer unto the said Corybant, Inc., its successors, legal representatives and assigns, our entire rights, title, and interests in and to the said invention and the aforesaid patent application, for the territory of the United States of America and for all foreign countries and to all Letters Patent, continuations, reissues, and extensions to be obtained therefore; and we further agree to cooperate with the assignee hereunder in the obtaining and sustaining of any and all such Letters patent, but at the expense of said assignee.
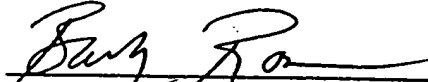
We do further agree to execute all applications for patent, assignments, and other appropriate documents and to perform all acts and to do all things necessary to make this Agreement effective and to perfect all right, title and interest in and to said discoveries, improvements or inventions in Corybant, Inc., its successors, legal representative and assigns, as the assignee of the entire right, title
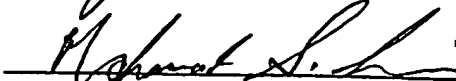
and interest therein.

IN WITNESS WHEREOF, we have hereunto set our hands and affixed our seals on the date set forth hereinafter.

Ronald M. Barber
Ronald M. Barber

Barbara J. Rossner
Barbara J. Rossner

Mohammad S. Salim
Mohammad S. Salim

STATE OF COLORADO   )
)ss.
COUNTY OF BOULDER   )

Before me, a Notary Public in and for the said County and State, personally appeared RONALD M. BARBER, BARBARA J. ROSSNER, and MOHAMMAD S. SALIM, known to be the persons whose names are subscribed to the foregoing instrument, and each acknowledged to me that he executed the same for the purposes and consideration therein expressed.

Given under my hand and seal of office this _8th_ day of _January_ , 2000.

(SEAL)

Jessica DeNovellis
Notary Public

My Commission Expires: _06/19/01_

2

VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(c)--SMALL BUSINESS CONCERN	Docket Number (optional) 19382-002

Applicants or Patentees: **RONALD M. BARBER ET AL.**
Serial or Patent No.:   **Not Yet Accorded**
Filed or Issued: **Not Yet Accorded**
Title: **DATASOURCE HARMONIZER**

I hereby declare that I am

[ ] the owner of the small business concern identified below:
[X] an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN: Corybant, Inc.
ADDRESS OF SMALL BUSINESS CONCERN: 3800 Arapahoe, Suite #205, Boulder, CO 80303

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

[X]   the specification filed herewith with title as listed above.
[ ]   the application identified above.
[ ]   the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights in the invention must file separate verified statements averring to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would qualify as an independent inventor under 37 CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization having any rights in the invention is listed below:
[X] no such person, concern, or organization exists.
[ ]   each person, concern or organization is listed below.

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING: Ronald M. Barber
TITLE OF PERSON IF OTHER THAN OWNER: President and CEO
ADDRESS OF PERSON SIGNING: 3800 Arapahoe, Suite #205, Boulder, CO 80303

SIGNATURE _____   DATE   1/19/00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Barber et al.	)
		) Group Art Unit:
Serial No.:	Not Yet Accorded	) Not Yet Assigned
		)
Filing Date:	Not Yet Accorded	) Examiner:
		) Not Yet Assigned
Title:	DATASOURCE HARMONIZER	)
		)
Our File No.:	19382.002	)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INTERVENTION BY ASSIGNEE, REVOCATION OF POWER OF ATTORNEY

AND ADDRESS FOR CORRESPONDENCE

To: Assistant Commissioner for Patents
       Washington, D.C.   20231

Sir:

Intervention and Revocation

Corybant, Inc., a corporation organized and existing under the laws of the State of Colorado, the assignee of record of the entire right, title, and interest in and to the invention and the above-referenced patent application, **hereby intervenes** in this patent application and **revokes all powers of attorney** previously appointed by the inventor or by any other entity in this patent application.

Appointment of Power of Attorney
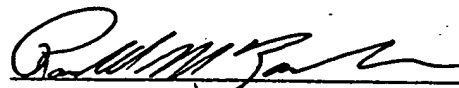
Corybant, Inc. hereby appoints, effective immediately, as principal attorneys: James R.

Young, Reg. No. 27,847; Thomas C. Folsom, Reg. No. 35,514; Steven C. Petersen, Reg. No.

36,238; Kent A. Lembke, Reg. No. 44,866; and agent Sarah S. O'Rourke, Reg. No. 41,226.

Address for Correspondence

Please direct all communications to the following address:

Thomas C. Folsom
CHRISMAN, BYNUM & JOHNSON, P.C.
1900 Fifteenth Street
Boulder, Colorado 80302
Telephone: (303) 546-1300
Fax: (303) 449-5426

I hereby declare that all statements made herein of my own knowledge are true and that

all statements made on information and belief are believed to be true; and further that these

statements were made with the knowledge that willful false statements and the like so made are

punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States

Code and that such willful false statements may jeopardize the validity of the application or any

patent issued thereon.

Ronald M. Barber, President and CEO

Date: 1/19/00

CERTIFICATE UNDER 37 CFR 3.73(b)

Applicant(s):   Barber et al.

Application No.: Not Yet Accorded   Filed: Not Yet Accorded

Entitled: DATASOURCE HARMONIZER

Corybant, Inc.   , a corporation
(Name of Assignee)   (Type of Assignee, e.g. corporation, partnership, university, government agency, etc.)

certifies that it is the assignee of the entire right, title and interest in the patent application identified above by virtue of either:

A. [X]   An assignment from the inventor(s) of the patent application identified above. The assignment was recorded in the Patent and
         Trademark Office at Reel _____, Frame _____, or for which a copy is attached.
OR

B. [ ]   A chain of title from the inventor(s) of the patent application identified above, to the current assignee as shown below:

    1.   From:_____ To:_____
    The document was recorded in the Patent and Trademark Office at
           Reel _____, Frame _____, or for which a copy thereof is attached.

    2.   From:_____ To:_____
    The document was recorded in the Patent and Trademark Office at
           Reel _____, Frame _____, or for which a copy thereof is attached.

    3.   From:_____ To:_____
    The document was recorded in the Patent and Trademark Office at
           Reel _____, Frame _____, or for which a copy there is attached.

    [ ]   Additional documents in the chain of title are listed on a supplemental sheet.

[X] Copies of assignment or other documents in the chain of title are attached.

The undersigned has reviewed all the documents in the chain of title of the patent application identified above and, to the best of undersigned's knowledge and belief, title is in the assignee identified above.

The undersigned (whose title is supplied below) is empowered to sign this certificate on behalf of the assignee.

I hereby declare that all statements made herein of my own knowledge are true, and that all statements made on information and belief are believed to be true; and further, that these statements are made with the knowledge that willful false statements, and the like so made, are punishable by fine or imprisonment, or both, under Section 1001, Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

1/19/00
Date

Signature

Ronald M. Barber
Typed or printed name

President and CEO, Corybant Inc.
Title

harmeng

Logical View Report

Unified Modeling Language Syntax

Includes Attributes And Operations

Generated April 14, 1999

9:12:55 AM

E:\Corybant\HarmEng\$design\harmeng.mdl

TABLE OF CONTENTS

LOGICAL VIEW REPORT

Logical View

harmeng

CAction

Public Operations:

CAction (pHamUID : CHamUID* = NULL) : CAction
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
~CAction () :
Execute (lpszPatientContext : LPCTSTR, pCallback : FARPROC) : HRESULT
ExecuteDone (ppRowSet : BSTR*, pRowsAffected : ULONG*) : HRESULT
AddMessage (pMsgObj : CMessage*) : void
RemoveMessage (pMsgObj : CMessage*) : void
RemoveMessage (MessageUID : CHamUID&) : void
GetStartMessage () : POSITION
GetNextMessage (pos : POSITION&) : CHamUID*
GetMessage () : CHamUID*
GetFirstMessage (rCathamUID : CHamUID&) : CHamUID*
GetMessageUIDList () : CHamUIDList*
GetSpec () : CSpecObject*

Private Operations:

DeSerializeVer1 (ar : CArchive&) : void

CIAction

Derived from [unnamed]

Public Operations:

CIAction () : CIAction
get_UID (pVal : BSTR*) : HRESULT
get_Description (pVal : BSTR*) : HRESULT
put_Description (newVal : BSTR) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
put_Name (newVal : BSTR) : HRESULT
get_Commands (pVal : IDispatch**) : HRESULT
AddCommand (CmdName : BSTR) : HRESULT
RemoveCommand (CmdName : BSTR) : HRESULT
SetAction (pAction : CAction*) : void
GetAction () : CAction*

LOGICAL VIEW REPORT

harmeng

CAction

Public Operations:

CAction (pHamUID : CHamUID* = NULL) : CAction
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
~CAction () :
Execute (lpszPatientContext : LPCTSTR, pCallback : FARPROC) : HRESULT
ExecuteDone (ppRowSet : BSTR*, pRowsAffected : ULONG*) : HRESULT
AddMessage (pMsgObj : CMessage*) : void
RemoveMessage (pMsgObj : CMessage*) : void
RemoveMessage (MessageUID : CHamUID&) : void
GetStartMessage () : POSITION
GetNextMessage (pos : POSITION&) : CHamUID*
GetMessage () : CHamUID*
GetFirstMessage (rCathamUID : CHamUID&) : CHamUID*
GetMessageUIDList () : CHamUIDList*
GetSpec () : CSpecObject*

Private Operations:

DeSerializeVer1 (ar : CArchive&) : void

CIAction

Derived from [unnamed]

Public Operations:

CIAction () : CIAction
get_UID (pVal : BSTR*) : HRESULT
get_Description (pVal : BSTR*) : HRESULT
put_Description (newVal : BSTR) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
put_Name (newVal : BSTR) : HRESULT
get_Commands (pVal : IDispatch**) : HRESULT
AddCommand (CmdName : BSTR) : HRESULT
RemoveCommand (CmdName : BSTR) : HRESULT
SetAction (pAction : CAction*) : void
GetAction () : CAction*

LPCATEGORY
MULTIPLICITY
eOPTIONAL
DATASOURCE
TTErrorHandler

Derived from TTResultCode

Public Operations:

TTErrorHandler () : TTErrorHandler
TTErrorHandler () : TTErrorHandler
bErrorHandlerInit () : BOOL
vSetInstance () : void
~TTErrorHandler () :
vSetSeverityThreshold (nLevel : SEVERITY_LEVEL = TTER_CRITICAL) : void
bIsLoggingOn (: void) : BOOL
nGetErrSource () : UINT
vRunSilent (bSilent : BOOL) : void
bIsSilent (: void) : BOOL
nGetSeverityThreshold () : SEVERITY_LEVEL
bLogFilename (: const char*) : BOOL
szGetLogFilename (: void) : CString
hLoggingOn (: void) : TTRESULT
vLoggingOff (: void) : void
iMessageBox (: TTRESULT, fuStyle : UINT = NULL) : int
iMessageBox (szMessage : CString&, fuStyle : UINT = NULL) : int
hWriteToLog (szMessage : CString&) : TTRESULT

Private Operations:

bIsErrorMessage () : BOOL
nGetSeverityIcon () : UINT
bIsSeverityOn () : BOOL
szMessageFilter (nMessageID : UINT) : CString
GetLogFilename (szLogFilename : CString&) : void
hOpenLogFile (: void) : TTRESULT
CloseLogFile (: void) : void
bLogFileSizeTooBig (: void) : BOOL

PCTTERRORHANDLER
LPCTTERRORHANDLER
(CProxy_IExecuteEvents<CExecute>)
CProxy_IExecuteEvents
CHamRole

Derived from [unnamed], IHamRole

Public Operations:

CHamRole () : CHamRole
GetCategory (strName : BSTR, pCategory : void**) : HRESULT
GetSpec (pRole : void**) : HRESULT
GetCatDS (strName : BSTR, pDS : void**) : HRESULT
GetActionList (pLst : void**) : HRESULT

HAMUID
PHAMUID
CParameter

Public Operations:

CParameter (iParam : DWORD) : CParameter
Index () : DWORD
FToVariant (varDest : VARIANT&) : BOOL
~CParameter () :

CByValParameter

Derived from CParameter

Public Operations:

CByValParameter (iParam : DWORD, szParam : LPTSTR) : CByValParameter
FToVariant (varDest : VARIANT&) : BOOL
~CByValParameter () :

CParser

Protected Attributes:

m_cBindableParams : int
m_cByValParams : int

Public Operations:

HasRetval () : BOOL
CParamsWORetval () : int
GetMethod () : LPCTSTR
GetTable () : LPTSTR
CParser (pCommand : CHarmEngCommand* = NULL) : CParser
~CParser () :
CommandType () : enum COMMANDTYPE
FParseSQLCall (szSQL : LPCTSTR) : BOOL
CBindableParameters () : int
ConvInsert2Send (szSQL : CString&) : BOOL
ConvInsert2Print (szSQL : CString&) : BOOL

Protected Operations:

FInit () : BOOL
RemoveWhiteSpace (sz : LPCTSTR, ich : int&) : void
FindNextWhiteSpace (sz : LPCTSTR, ich : int&) : void
FindEndParam (sz : LPCTSTR, ich : int&) : void
FindEndMethod (sz : LPCTSTR, ich : int&, fIsQuoted : BOOL) : void
FindEndCommand (sz : LPCTSTR, ich : int&, fIsQuoted : BOOL) : void
FParseCallParams (sz : LPCTSTR, ich : int&, fCallOpen : BOOL) : BOOL
FCopyByValParam (sz : LPCTSTR, ich : int&) : BOOL
FindMethod (szSQL : LPCTSTR, ich : int) : BOOL
FindCommand (szSQL : LPCTSTR, ich : int) : BOOL
SelectCommand (szSQL : LPCTSTR, ich : int) : BOOL
InsertCommand (szSQL : LPCTSTR, ich : int) : BOOL
PrintCommand (szSQL : LPCTSTR, ich : int) : BOOL
SendCommand (szSQL : LPCTSTR, ich : int) : BOOL
FaxCommand (szSQL : LPCTSTR, ich : int) : BOOL
ConvInsert2 (szSQL : CString&, szCmd : LPCTSTR) : BOOL
ExtractTableName (szSQL : LPCTSTR, ich : int) : void

COMMANDTYPE
RULEBIND

Public Operations:

RULEBIND () : RULEBIND

CRule

Public Operations:

CRule (pHamUID : CHamUID* = NULL) : CRule
Draw (pdc : CDC*) : void
~CRule () :
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
GetSpec () : CSpecObject*

Private Operations:

DeSerializeVer1 (ar : CArchive&) : void
DeSerializeVer2 (ar : CArchive&) : void

CIRule

Derived from [unnamed], ISupportErrorInfo

Public Operations:

CIRule () : CIRule
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get_UID (pVal : BSTR*) : HRESULT
get_Description (pVal : BSTR*) : HRESULT
put_Description (newVal : BSTR) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
put_Name (newVal : BSTR) : HRESULT
get_Source (pVal : BSTR*) : HRESULT
put_Source (newVal : BSTR) : HRESULT
get_SourceCategory (pVal : BSTR*) : HRESULT
put_SourceCategory (newVal : BSTR) : HRESULT
get_Format (pVal : BSTR*) : HRESULT
put_Format (newVal : BSTR) : HRESULT
get_BaseDataType (pVal : long*) : HRESULT
put_BaseDataType (newVal : long) : HRESULT
SetRule (pRule : CRule*) : void
GetRule () : CRule*

RolePackage

CRole

Derived from [unnamed], ISupportErrorInfo

Public Operations:

CRole () : CRole
~CRole () :
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
Open (pFilename : BSTR) : HRESULT
New (pFilename : BSTR) : HRESULT
Save () : HRESULT
Close () : HRESULT
get_UID (pVal : BSTR*) : HRESULT
get_Description (pVal : BSTR*) : HRESULT
put_Description (newVal : BSTR) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
put_Name (newVal : BSTR) : HRESULT
get_Categories (pVal : IDispatch**) : HRESULT
get_Rules (pVal : IDispatch**) : HRESULT
get_Actions (pVal : IDispatch**) : HRESULT
get_Associations (pVal : IDispatch**) : HRESULT
get_Commands (pVal : IDispatch**) : HRESULT

CSpecObject

Public Operations:

CSpecObject (pHamUID : CHamUID* = NULL) : CSpecObject
Draw (pdc : CDC*) : void
Draw (pParentWnd : CView*) : void
~CSpecObject () :
Serialize (ar : CArchive&) : void
DeSerializeVer1 (ar : CArchive&) : void
GetCurCategory () : CCategory*
SetCurCategory (ptr : CCategory*) : void
CategoryList () : CCategoryList*
Execute (lpszActionID : LPCTSTR, lpszPatientContext : LPCTSTR, pCallBack : FARPROC) : HRESULT
OpenRefFile (pstrRefDocFile : LPCOLESTR) : HRESULT
OpenRefFile (pstrRefDocFile : LPCTSTR) : HRESULT
ImportRefFile (pstrRefDocFile : LPCOLESTR) : HRESULT
SaveRefFile () : HRESULT
CloseRefFile () : HRESULT
GetRefFilePath () : CString
SetRefFilePath (pPath : LPCTSTR) : void
IsSameRefFile (pPath : LPCTSTR) : BOOL
RuleList () : CRuleList*
MessageLst () : CMessageList*
ActionLst () : CActionList*
ASSOCList () : CAssociationList*
GetUIDHelper () : CHamUIDHelper*

CExecute

Derived from [unnamed], ISupportErrorInfo, (CProxy_IExecuteEvents<CExecute>)

Public Attributes:

m_pUnkMarshaler : CComPtr
CALLBACK : int

Public Operations:

CExecute () : CExecute
~CExecute () :
FinalConstruct () : HRESULT
FinalRelease () : void
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
Action (pstrActionID : BSTR) : HRESULT
OpenRefFile (pstrRefDocFile : BSTR) : HRESULT
get_Context (pVal : BSTR*) : HRESULT
put_Context (newVal : BSTR) : HRESULT
get_TestContext (pVal : BSTR*) : HRESULT
put_TestContext (newVal : BSTR) : HRESULT
NotifyActionDone () : int

CategoryPackage

CCategories

Derived from [unnamed]

Private Attributes:

m_nSize : int

Public Operations:

CCategories () : CCategories
~CCategories () :
Add (InitStr : BSTR, Table : BSTR, pVal : VARIANT*) : HRESULT
Find (name : BSTR, pVal : VARIANT*) : HRESULT
get__NewEnum (pVal : IUnknown**) : HRESULT
get_Item (Index : long, pVal : VARIANT*) : HRESULT
get_Count (pVal : long*) : HRESULT
NoWorkAdd (pVal : ICategory*) : HRESULT
Delete (pName : BSTR) : HRESULT
Init (Spec : CSpecObject*, pCatLst : CCategoryList*) : HRESULT

CCategoryList

Public Operations:

CCategoryList () : CCategoryList
Draw (pdc : CDC*) : void
~CCategoryList () :
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
Add (pSpecObject : CSpecObject*, pOutCategory : CCategory**, pDSNStr : LPCTSTR = NULL, pstrTable :
    LPCTSTR = NULL) : HRESULT
Add (pSpec : CSpecObject*, pCategory : CCategory*) : HRESULT
DeSerializeVer1 (ar : CArchive&, pSpec : CSpecObject*) : void
DeleteAll () : void
Find (pName : LPCTSTR, pIndex : int* = NULL) : CCategory*
Find (hamUID : CHamUID, pIndex : int* = NULL) : CCategory*
New (ptr : CElementBase*) : CElementBase*
AddHead (pNewList : CCategory*) : POSITION
AddTail (pNewList : CCategory*) : POSITION
GetNext (rPosition : POSITION&) : CCategory*&
GetNext (rPosition : POSITION&) : CCategory*
GetPrev (rPosition : POSITION&) : CCategory*&
GetPrev (rPosition : POSITION&) : CCategory*

CICategory

Derived from [unnamed], ISupportErrorInfo

Public Operations:

CICategory () : CICategory
GetCategory () : CCategory*
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get__Category (pVal : long**) : HRESULT
get_IsVisible (pVal : BOOL*) : HRESULT
put_IsVisible (newVal : BOOL) : HRESULT
get_IsReferenceCategory (pVal : BOOL*) : HRESULT
put_IsReferenceCategory (newVal : BOOL) : HRESULT
get_ReferenceCategoryUID (pVal : BSTR*) : HRESULT
put_ReferenceCategoryUID (newVal : BSTR) : HRESULT
get_TableName (pVal : BSTR*) : HRESULT
put_TableName (newVal : BSTR) : HRESULT
get_Provider (pVal : BSTR*) : HRESULT
put_Provider (newVal : BSTR) : HRESULT
get_DataSource (pVal : BSTR*) : HRESULT
put_DataSource (newVal : BSTR) : HRESULT
get_DBInitStr (pVal : BSTR*) : HRESULT
put_DBInitStr (newVal : BSTR) : HRESULT
get_UID (pVal : BSTR*) : HRESULT
get_Description (pVal : BSTR*) : HRESULT
put_Description (newVal : BSTR) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
put_Name (newVal : BSTR) : HRESULT
get_EMailOnInsert (pVal : BOOL*) : HRESULT
put_EMailOnInsert (newVal : BOOL) : HRESULT
get_PrintOnInsert (pVal : BOOL*) : HRESULT
put_PrintOnInsert (newVal : BOOL) : HRESULT
get_ReadOnly (pVal : BOOL*) : HRESULT
put_ReadOnly (newVal : BOOL) : HRESULT
get_Elements (... : <any>) : HRESULT
SetCatUID (uid : CHamUID&) : void
GetCatUID () : CHamUID&
SetCategory (pCat : CCategory*) : void
CreateCategory () : HRESULT

CCategory

Public Operations:

CCategory (pHamUID : CHamUID* = NULL) : CCategory
CCategory (right : CCategory&) : CCategory
Draw (pdc : CDC*) : void
Draw (pParentWnd : CView*) : void
~CCategory () :
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
GetProcedures (pStrlst : CStringList*) : HRESULT
Add (pSpecObject : CSpecObject*) : HRESULT
Add (bRepair : BOOL = FALSE) : HRESULT
ElementList () : CElementList*
GetSpec () : CSpecObject*
MapTo (pCategory : CCategory*) : void
MakeAssociations () : HRESULT
GetActualTableName () : LPCTSTR
GetActualTableNameLength () : ULONG
GetTableName () : LPCTSTR
SetTableName (ptr : LPCTSTR) : void
GetTableNameLength () : ULONG
IsVisibleThruOLEDB () : BOOL
SetVisibleThruOLEDB (bVal : BOOL) : void
IsPrintInsert () : BOOL
IsEMailInsert () : BOOL
IsReadOnly () : BOOL
SetPrintInsert (bVal : BOOL) : void
SetEMailInsert (bVal : BOOL) : void
SetReadOnly (bVal : BOOL) : void
ReleaseDataSource () : void
SetDataSource () : HRESULT
GetDataSource () : COleDBDataSource*
GetDataSourceName () : CString
SetDataSourceName (ptr : LPCTSTR) : void
GetProvider () : CString
SetProvider (pszProvider : LPCTSTR) : void
IsReferenceCat () : BOOL
SetReferenceCat (bVal : BOOL) : void
GetReferenceCatName () : LPCTSTR
GetReferenceCatUID () : CHamUID&
SetReferenceCatUID (rVal : CHamUID&) : void
GetEmailAddress () : CString
SetEmailAddress (ptr : LPCTSTR) : void
GetInitializationString () : LPCTSTR
SetInitializationString (ptr : LPCTSTR) : void

Private Operations:

DeSerializeVer1 (ar : CArchive&) : void
DeSerializeVer2 (ar : CArchive&) : void
DeSerializeVer3 (ar : CArchive&) : void
DeSerializeVer4 (ar : CArchive&) : void
DeSerializeVer5 (ar : CArchive&) : void
DeSerializeVer6 (ar : CArchive&) : void
GetColSchemaInfo (bRepair : BOOL) : HRESULT
GetTableSchemaInfo (pList : CStringList*) : HRESULT

ElementPackage

HL7Optionality
CElements

Derived from [unnamed]

Private Attributes:

m_nSize : int

Public Operations:

CElements () : CElements
~CElements () :
Find (name : BSTR, pVal : VARIANT*) : HRESULT
get__NewEnum (pVal : IUnknown**) : HRESULT
get_Item (Index : long, pVal : VARIANT*) : HRESULT
get_Count (pVal : long*) : HRESULT
Add (Name : BSTR, pVal : VARIANT*) : HRESULT
Delete (pName : BSTR) : HRESULT
Init (Cat : CCategory*, pElementLst : CElementList*) : HRESULT

CElement

Derived from [unnamed], ISupportErrorInfo

Public Operations:

CElement () : CElement
~CElement () :
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get_UID (pVal : BSTR*) : HRESULT
get_Description (pVal : BSTR*) : HRESULT
put_Description (newVal : BSTR) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
put_Name (newVal : BSTR) : HRESULT
get_Ordinal (pVal : long*) : HRESULT
put_Ordinal (newVal : long) : HRESULT
get_Precision (pVal : BYTE*) : HRESULT
put_Precision (newVal : BYTE) : HRESULT
get_Scale (pVal : BYTE*) : HRESULT
put_Scale (newVal : BYTE) : HRESULT
get_Flags (pVal : long*) : HRESULT
put_Flags (newVal : long) : HRESULT
get_Size (pVal : long*) : HRESULT
put_Size (newVal : long) : HRESULT
get_Type (pVal : long*) : HRESULT
put_Type (newVal : long) : HRESULT
get_TypeStr (pVal : BSTR*) : HRESULT
get_Association (... : <any>) : HRESULT
SetElement (pElement : CDataElement*) : void
GetElement () : CDataElement*

Private Operations:

Clean () : void

CDataElement

Public Operations:

CDataElement (pHamUID : CHamUID* = NULL) : CDataElement
CDataElement (: CDataElement&) : CDataElement
CDataElement (: ATLCOLUMNINFO&) : CDataElement
operator = (: const DBCOLUMNINFO&) : const CDataElement&
operator = (: const ATLCOLUMNINFO&) : const CDataElement&
Draw (pdc : CDC*) : void
~CDataElement () :
Serialize (ar : CArchive&, pCat : CCategory*) : void
MakeAssociations () : HRESULT
GetColumnEntry (pColInfo : ATLCOLUMNINFO*, nCurOffset : ULONG&) : BOOL
GetColumnEntry (pColInfo : DBCOLUMNINFO*) : BOOL
GetDBTYPE () : DBTYPE
GetDataHolderLength () : ULONG
GetDataHolder () : LPCTSTR
SetDataHolder (ptr : LPCTSTR) : void
GetAssociation () : CAssociation*
GetAssociationUID () : CHamUID*
SetAssociationUID (uid : CHamUID&) : void
GetSpec () : CSpecObject*
GetOptionality () : enum HL7Optionality
SetOptionality (val : enum HL7Optionality) : void
CanRepeat () : BOOL
SetCanRepeat (val : BOOL) : void
GetRepetitionCnt () : ULONG
SetRepetitionCnt (val : ULONG) : void
GetFormatRuleUID () : CHamUID*
SetFormatRuleUID (val : CHamUID&) : void

Private Operations:

DeSerializeVer1 (ar : CArchive&) : void
DeSerializeVer2 (ar : CArchive&) : void
DeSerializeVer3 (ar : CArchive&) : void
DeSerializeVer4 (ar : CArchive&) : void

CommandPackage

CMessage

Public Operations:

CMessage (pHamUID : CHamUID* = NULL) : CMessage
Draw (pdc : CDC*) : void
~CMessage () :
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
IsValid () : BOOL
IsCategoryValid () : BOOL
IsMessagePartValid () : BOOL
ReplaceCategory (pCategory : CCategory*) : void
Execute (: CAction*, lpszPatContext : LPCTSTR, ppRowSet : BSTR*, pRowsAffected : ULONG*) :
    HRESULT
AddToMessage (pCategory : CCategory*, pSpecListUID : CHamUID*, ndsCommand : enum
    SQLMessageParts, StringList : CStringList&, szSQLScript : CString&, szContext : CString&) : void
InitMessage (pCategory : CCategory*) : HRESULT
AddSelectToMessage (StringList : CStringList&) : HRESULT
AddWhereToMessage (StringList : CStringList&, szContext : LPCTSTR) : HRESULT
AddCallToMessage (szSQLScript : CString&) : HRESULT
AddInsertToMessage (StringList : CStringList&, ndsCommand : enum SQLMessageParts = msgINSERT) :
    HRESULT
GetSpec () : CSpecObject*
GetMsgPartsList () : CMessageList*
IsProcedure () : BOOL
SetIsProcedure (state : BOOL) : void

Private Operations:

DeSerializeVer1 (ar : CArchive&, pSpec : CSpecObject*) : void
DeSerializeVer2 (ar : CArchive&, pSpec : CSpecObject*) : void
DeSerializeVer3 (ar : CArchive&, pSpec : CSpecObject*) : void
DeSerializeVer4 (ar : CArchive&, pSpec : CSpecObject*) : void
MakeInsertMessage (pCategory : CCategory*, pSpecListUID : CHamUID*, pos : POSITION, StringList :
    CStringList&, szTmpString : CString&, szdsCommand : CString&) : void
GetData (pCommand : LPCTSTR, pnRowsEffected : ULONG*, pCat : CCategory*) : HRESULT
InternalGetData (rs : CCommand&, pCategory : CCategory*, pnRowsEffected : ULONG*) : HRESULT
InternalGetData (Session : CSession&, pszCommand : LPCTSTR, pCategory : CCategory*, pnRowsEffected :
    ULONG*) : HRESULT
SetData (pCommand : LPCTSTR, pnRowsEffected : ULONG*, pCat : CCategory*) : HRESULT
InternalSetData (Session : CSession&, pszCommand : LPCTSTR, rs : CCommand&, pCategory :
    CCategory*, pnRowsEffected : ULONG*) : HRESULT
AddMessagePart (pMsgObj : CMessage*) : void
RemoveMessagePart (pMsgObj : CMessage*) : void
GetStartMessagePart () : POSITION
GetNextMessagePart (pos : POSITION&) : CMessage*
GetMessagePart () : CMessage*
GetFirstMessagePart (rCathamUID : CHamUID&) : CMessage*
HasWildCard () : BOOL
FixMessages () : void

CMessageList

Public Operations:

CMessageList () : CMessageList
Draw (pdc : CDC*) : void
~CMessageList () :
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
DeSerializeVer1 (ar : CArchive&, pSpec : CSpecObject*) : void
FixMessages () : void
DeleteAll () : void
Find (pName : LPCTSTR, pIndex : int* = NULL) : CMessage*
Find (rUID : CHamUID&, pIndex : int* = NULL) : CMessage*
DeleteMessage (rMsgUID : CHamUID&) : void
New (ptr : CElementBase*) : CElementBase*
AddHead (pNewList : CMessage*) : POSITION
AddTail (pNewList : CMessage*) : POSITION
GetNext (rPosition : POSITION&) : CMessage*&
GetNext (rPosition : POSITION&) : CMessage*
GetPrev (rPosition : POSITION&) : CMessage*&
GetPrev (rPosition : POSITION&) : CMessage*

CdsCommand

Derived from [unnamed], ISupportErrorInfo

Public Operations:

CdsCommand () : CdsCommand
InitMessage (pCategoyUID : BSTR) : HRESULT
AddSelectToMessage (ElementName : BSTR) : HRESULT
AddWhereToMessage (ElementName : BSTR, szContext : BSTR) : HRESULT
AddCallToMessage (szProcedure : BSTR) : HRESULT
AddInsertToMessage (ElementName : BSTR) : HRESULT
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get_UID (pVal : BSTR*) : HRESULT
get_Description (pVal : BSTR*) : HRESULT
put_Description (newVal : BSTR) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
put_Name (newVal : BSTR) : HRESULT
get_CommandParts (pVal : IDispatch**) : HRESULT
get_IsKeyWord (pVal : BOOL*) : HRESULT
put_IsKeyWord (newVal : BOOL) : HRESULT
get_KeyWord (pVal : BSTR*) : HRESULT
put_KeyWord (newVal : BSTR) : HRESULT
SetMessage (pMessage : CMessage*) : void
GetMessage () : CMessage*

SQLMessageParts
CdsCommands

Derived from [unnamed]

Private Attributes:

m_nSize : int

Public Operations:

CdsCommands () : CdsCommands
~CdsCommands () :
Find (name : BSTR, pVal : VARIANT*) : HRESULT
get__NewEnum (pVal : IUnknown**) : HRESULT
get_Item (Index : long, pVal : VARIANT*) : HRESULT
get_Count (pVal : long*) : HRESULT
Add (pName : BSTR, pVal : VARIANT*) : HRESULT
Delete (pName : BSTR) : HRESULT
Init (Spec : CSpecObject*, pMsgLst : CMessageList*) : HRESULT

Utility Package

CHamUID

Public Operations:

CHamUID () : CHamUID
IsSpecValid () : BOOL
IsSpecListValid () : BOOL
IsCategoryValid () : BOOL
IsElementValid () : BOOL
IsValid () : BOOL
IsSpecListUID () : BOOL
SetInvalid () : void
ReplaceCategory (rhamuid : const CHamUID&) : void
ReplaceElement (rhamuid : const CHamUID&) : void
GetSpec () : CSpecObject*
GetCategory () : CCategory*
GetElement () : CElementBase*
UpdateSpec (pElem : CSpecObject*) : BOOL
UpdateCategory (pElem : CCategory*) : BOOL
UpdateElement (pElem : CElementBase*) : BOOL
Convert () : BOOL
operator LPCTSTR () : LPCTSTR
operator << (ar : CArchive&, hamUID : CHamUID) : int
operator >> (ar : CArchive&, rhamUID : CHamUID&) : int
operator = (s2 : const CHamUID&) : const CHamUID&

CElementBase

Derived from CObject

Public Operations:

CElementBase () : CElementBase
CElementBase (right : CElementBase&) : CElementBase
Draw (: CDC*) : void
~CElementBase () :
Serialize (ar : CArchive&) : void
DeSerializeVer1 (ar : CArchive&) : void
GetName () : LPCTSTR
SetName (szname : LPCTSTR) : void
GetDescription () : LPCTSTR
SetDescription (szdesc : LPCTSTR) : void
GetUID () : CHamUID&
IsRemoved () : BOOL
IsDeleted () : BOOL
SetRemoved (state : BOOL) : void
SetDeleted (state : BOOL) : void

CElementList

Public Operations:

CElementList () : CElementList
Draw (pdc : CDC*) : void
~CElementList () :
Serialize (ar : CArchive&, pCat : CCategory* = NULL) : void
DeSerializeVer1 (ar : CArchive&, pCat : CCategory*) : void
DeleteAll () : void
New (ptr : CElementBase*) : CElementBase*
Find (pName : LPCTSTR, pIndex : int* = NULL) : CDataElement*
Find (hamUID : CHamUID, pIndex : int* = NULL) : CDataElement*
GetSortedList (ASSOCList : CAssociationList*) : CElementList*
SortListByOrdinal () : void
AddHead (pNewList : CDataElement*) : POSITION
AddTail (pNewList : CDataElement*) : POSITION
GetNext (rPosition : POSITION&) : CDataElement*&
GetNext (rPosition : POSITION&) : CDataElement*
GetPrev (rPosition : POSITION&) : CDataElement*&
GetPrev (rPosition : POSITION&) : CDataElement*

CHamUIDList

Derived from CObList

Public Operations:

~CHamUIDList () :
New (ptrParent : CHamUID*) : CHamUID*
DeleteAll () : void
AddHead (pNewList : CHamUID*) : POSITION
AddTail (pNewList : CHamUID*) : POSITION
GetNext (rPosition : POSITION&) : CHamUID*&
GetNext (rPosition : POSITION&) : CHamUID*
GetPrev (rPosition : POSITION&) : CHamUID*&
GetPrev (rPosition : POSITION&) : CHamUID*

CHamUIDs

Derived from [unnamed], ISupportErrorInfo

Private Attributes:

m_nSize : int

Public Operations:

CHamUIDs () : CHamUIDs
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get__NewEnum (pVal : IUnknown**) : HRESULT
get_Item (Index : long, pVal : VARIANT*) : HRESULT
get_Count (pVal : long*) : HRESULT
Init (UIDList : CHamUIDList*) : HRESULT

CHamUIDHelper

Private Attributes:

m_arSpec : CPtrList
m_arSpecList : CPtrList
m_arCategory : CPtrList
m_arElement : CPtrList

Public Operations:

CHamUIDHelper () : CHamUIDHelper
UpdateSpec (pElem : CElementBase*, ruid : CHamUID&) : BOOL
UpdateCategory (rparentuid : const CHamUID&, pElem : CElementBase*, ruid : CHamUID&) : BOOL
UpdateElement (rparentuid : const CHamUID&, pElem : CElementBase*, ruid : CHamUID&) : BOOL
CreateSpec (phamuid : const CHamUID*, pSpec : CElementBase*, pNewuid : CHamUID*) : void
~CHamUIDHelper () :
CreateCategory (phamuid : const CHamUID*, pSpec : CElementBase*, pNewuid : CHamUID*) : void
CreateElement (rhamuid : const CHamUID&, pElem : CElementBase*, pNewuid : CHamUID*) : void
FindSpec (phamuid : const CHamUID*) : CSpecObject*
FindCategory (phamuid : const CHamUID*) : CCategory*
FindElement (phamuid : const CHamUID*) : CElementBase*

CIHamUID

Derived from [unnamed], ISupportErrorInfo

Public Operations:

CIHamUID () : CIHamUID
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get_UID (pVal : BSTR*) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
get_Valid (pVal : BOOL*) : HRESULT
SetUID (puid : CHamUID*) : void
GetUID () : CHamUID*

CElementBaseList

Derived from CObList

Public Operations:

~CElementBaseList () :
Serialize (ar : CArchive&) : void
New (ptrParent : CElementBase*) : CElementBase*
Find (pName : LPCTSTR, pIndex : int* = NULL) : CElementBase*
DeleteAll () : void
Delete (pSelObj : CElementBase*) : void
Delete (pUID : CHamUID*) : void
AddHead (pNewList : CElementBase*) : POSITION
AddTail (pNewList : CElementBase*) : POSITION
GetNext (rPosition : POSITION&) : CElementBase*&
GetNext (rPosition : POSITION&) : CElementBase*
GetPrev (rPosition : POSITION&) : CElementBase*&
GetPrev (rPosition : POSITION&) : CElementBase*

AssociationPackage

CAssociationList

Public Operations:

CAssociationList () : CAssociationList
Draw (pdc : CDC*) : void
~CAssociationList () :
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
DeSerializeVer1 (ar : CArchive&, pSpec : CSpecObject*) : void
DeleteAll () : void
Find (pName : LPCTSTR, pIndex : int* = NULL) : CAssociation*
Find (hamUID : CHamUID, pIndex : int* = NULL) : CAssociation*
Add (pElement1 : CDataElement*, pElement2 : CDataElement*, pOutAssoc : CAssociation**) : HRESULT
GetName (hamUID : CHamUID, szName : CString&) : BOOL
New (ptr : CElementBase*) : CElementBase*
AddHead (pNewList : CAssociation*) : POSITION
AddTail (pNewList : CAssociation*) : POSITION
GetNext (rPosition : POSITION&) : CAssociation*&
GetNext (rPosition : POSITION&) : CAssociation*
GetPrev (rPosition : POSITION&) : CAssociation*&
GetPrev (rPosition : POSITION&) : CAssociation*
TestAssociations () : BOOL

CIAssociations

Derived from [unnamed], ISupportErrorInfo

Private Attributes:

m_nSize : int

Public Operations:

```
CIAssociations () : CIAssociations
~CIAssociations () :
Find (name : BSTR, pVal : VARIANT*) : HRESULT
Add (Elem1UID : BSTR, Elem2UID : BSTR, pVal : VARIANT*) : HRESULT
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get__NewEnum (pVal : IUnknown**) : HRESULT
get_Item (Index : long, pVal : VARIANT*) : HRESULT
get_Count (pVal : long*) : HRESULT
Delete (pName : BSTR) : HRESULT
Init (Spec : CSpecObject*, pAssocLst : CAssociationList*) : HRESULT
```

CAssocParts

Derived from [unnamed], ISupportErrorInfo

Private Attributes:

```
m_nSize : int
```

Public Operations:

```
CAssocParts () : CAssocParts
~CAssocParts () :
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get__NewEnum (pVal : IUnknown**) : HRESULT
get_Item (Index : long, pVal : VARIANT*) : HRESULT
get_Count (pVal : long*) : HRESULT
Add (pVal : VARIANT*) : HRESULT
Delete (pName : BSTR) : HRESULT
Find (name : BSTR, pVal : VARIANT*) : HRESULT
Init (pAssocObject : CAssociation*) : HRESULT
```

CIAssociation

Derived from [unnamed], ISupportErrorInfo

Public Operations:

```
CIAssociation () : CIAssociation
AddAssocElement (strUID : BSTR) : HRESULT
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get_UID (pVal : BSTR*) : HRESULT
get_Description (pVal : BSTR*) : HRESULT
put_Description (newVal : BSTR) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
put_Name (newVal : BSTR) : HRESULT
get_AssocParts (pVal : IDispatch**) : HRESULT
get_Rule (pVal : IDispatch**) : HRESULT
SetAssoc (pAssoc : CAssociation*) : void
GetAssoc () : CAssociation*
```

CAssocPart

Derived from [unnamed]

Public Operations:

CAssocPart () : CAssocPart
get_UID (pVal : BSTR*) : HRESULT
get_Name (pVal : BSTR*) : HRESULT
get_Element (pVal : IDispatch**) : HRESULT
SetAssocPart (puid : CAssocDataElement*) : void
GetAssocPart () : CAssocDataElement*

CAssocDataElement

Public Operations:

CAssocDataElement () : CAssocDataElement
CAssocDataElement (right : CAssocDataElement&) : CAssocDataElement
CAssocDataElement (rDataElement : CDataElement&) : CAssocDataElement
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
~CAssocDataElement () :
GetDataElement () : CDataElement*
GetDataElementUID () : CHamUID*
GetCategoryUID () : CHamUID*

Private Operations:

DeSerializeVer1 (ar : CArchive&) : void
DeSerializeVer2 (ar : CArchive&) : void

CAssociation

Public Operations:

CAssociation (specUID : CHamUID* = NULL) : CAssociation
Draw (pdc : CDC*) : void
~CAssociation () :
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
DeSerializeVer1 (ar : CArchive&, pSpec : CSpecObject*) : void
FixupElement (pDataEle : CDataElement*) : void
AddAssocElement (pDataEle : CDataElement*) : void
RemoveAssocElement (pDataEle : CDataElement*) : void
RemoveAssocElement (pDataEleUID : CHamUID*) : void
GetStartAssocElement () : POSITION
GetNextAssocElement (pos : POSITION&) : CHamUID*
GetNextAssoc (pos : POSITION&) : CAssocDataElement*
GetAssocElement () : CAssocDataElement*
GetFirstAssocElement (rCathamUID : CHamUID&) : CAssocDataElement*
GetRule () : CRule*
SetRuleUID (rHamUID : CHamUID&) : void
GetRuleUID () : CHamUID
GetAssocElements () : CElementBaseList*
SetAssocElements (ptr : CElementBaseList*) : void
GetSpec () : CSpecObject*
TestAssociation () : BOOL

RulePackage

CRuleList

Public Operations:

CRuleList () : CRuleList
Draw (pdc : CDC*) : void
~CRuleList () :
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
DeSerializeVer1 (ar : CArchive&, pSpec : CSpecObject*) : void
DeleteAll () : void
Find (pName : LPCTSTR, pIndex : int* = NULL) : CRule*
Find (rHamUID : CHamUID&, pIndex : int* = NULL) : CRule*
New (ptr : CElementBase*) : CElementBase*
Import (pSpecObject : CSpecObject*) : HRESULT
GetData (pSpecObject : CSpecObject*, Session : CSession&, szTableName : LPCTSTR) : HRESULT
GetTableSchemaInfo (Session : CSession&, pList : CStringList*) : HRESULT
AddHead (pNewList : CRule*) : POSITION
AddTail (pNewList : CRule*) : POSITION
GetNext (rPosition : POSITION&) : CRule*&
GetNext (rPosition : POSITION&) : CRule*
GetPrev (rPosition : POSITION&) : CRule*&
GetPrev (rPosition : POSITION&) : CRule*

CRules

Derived from [unnamed], ISupportErrorInfo

Private Attributes:

m_nSize : int

Public Operations:

CRules () : CRules
~CRules () :
Find (name : BSTR, pVal : VARIANT*) : HRESULT
InterfaceSupportsErrorInfo (riid : REFIID) : HRESULT
get__NewEnum (pVal : IUnknown**) : HRESULT
get_Item (Index : long, pVal : VARIANT*) : HRESULT
get_Count (pVal : long*) : HRESULT
Add (Name : BSTR, pVal : VARIANT*) : HRESULT
Delete (pName : BSTR) : HRESULT
Init (Spec : CSpecObject*, pRuleLst : CRuleList*) : HRESULT

ActionPackage

CActionList

Public Operations:

CActionList () : CActionList
Serialize (ar : CArchive&, pSpec : CSpecObject*) : void
~CActionList () :
RemoveMessage (MessageUID : CHamUID&) : void
DeSerializeVer1 (ar : CArchive&, pSpec : CSpecObject*) : void
DeleteAll () : void
Draw (pdc : CDC*) : void
Find (pName : LPCTSTR, pIndex : int* = NULL) : CAction*
New (ptr : CElementBase*) : CElementBase*
AddHead (pNewList : CAction*) : POSITION
AddTail (pNewList : CAction*) : POSITION
GetNext (rPosition : POSITION&) : CAction*&
GetNext (rPosition : POSITION&) : CAction*
GetPrev (rPosition : POSITION&) : CAction*&
GetPrev (rPosition : POSITION&) : CAction*

CIActions

Derived from [unnamed]

Private Attributes:

m_nSize : int

Public Operations:

CIActions () : CIActions
~CIActions () :
Find (name : BSTR, pVal : VARIANT*) : HRESULT
get__NewEnum (pVal : IUnknown**) : HRESULT
get_Item (Index : long, pVal : VARIANT*) : HRESULT
get_Count (pVal : long*) : HRESULT
Add (Name : BSTR, pVal : VARIANT*) : HRESULT
Delete (pName : BSTR) : HRESULT
Init (Spec : CSpecObject*, pActionLst : CActionList*) : HRESULT

FARPROC
POSITION
CString
[unnamed]
CComVariant
ISupportErrorInfo
BOOL
COleDBDataSource
CSession
CDataSource
DBTYPE
ULONG
BYTE
DBCOLUMNFLAGS
UINT
ITypeInfo
DBID
CObject
CObList
HINSTANCE
SEVERITY_LEVEL
CFile
TTResultCode
IHamRole
LONG
DWORD
LPTSTR
CHarmEngCommand
TCHAR
LPSTORAGE

TOTALS:

12 Logical Packages
86 Classes

LOGICAL PACKAGE STRUCTURE

Logical View
  harmeng
      RolePackage
      CategoryPackage
      ElementPackage
      CommandPackage
      UtilityPackage
      AssociationPackage
      RulePackage
      ActionPackage
  <unspecified>

References

1   Feigenbaum and Feldman "Computers and Thought" McGraw Hill 1963 Library of Congress Catalog Number 63-17596 pp. 11-35 is a paper originally by A. M. Turing on Computing machinery and intelligence published in Mind, October 1950, 50:443-460

2   James L. Adams "Conceptual BlockBusting, a Guide to Better Ideas" Persus Books 1986 ISBN 0-201-55086-5 pp. 1-10, 83-101

3   Watts S. Humphrey "Managing the Software Process, SEI series in Software Engineering" Addison Wesley 1990 ISBN 0-201-18095-2 pp 72-80

4   Peter M. Senge "The Fifth Discipline" Doubleday/Currency 1990 ISBN: 0-385-26094-6 pp. 273-301

5   Smith and Reinertsen "Developing Products in Half the Time" Van Nostrand Reinhold 1991 ISBN 0-442-00243-2 pp. 3-6

6   Ron Burnt, Andrew Hutt "Open Framework, The System Architecture: An introduction" Prentice Hall 1992 ISBN 0-13-560186-x

7   Hamden-Turner and Trompenaars "The Seven Cultures of Capitalism" Currency Doubleday 1993 ISBN0-385-42101-x pp. 163-196

8   Lawrance M. Miller et al " Whole System Architecture: Beyond Reengineering: Designing the High Performance Organization" Miller Consulting Group Inc. 1994 ISBN:0-9629679-2-0 pp. 263-290

9   The Workflow management Coalition Specification Document Number TC00-1003 "The Workflow Reference Model" 29 Nov 1994

10 The Common Object Request Broker reference and specification Revision 2 July 1995

11 International standard CEI IEC 601-1-4 Part 4 Collateral Standard Programmable Electrical Medical Systems 1996

12 W3C Open Software Description Format (OSD) Submitted August 1997

13 Business Service Architecture, a NAC Position Paper April 1998

14 Microsoft OLEDB 2.0 Microsoft press 1998 ISBN 0-7356-0590

15 Jim Conallen "Building Web Applications with UML" Addison Wesley 1999 ISBN 00-201-61577-0 pp. 81-95